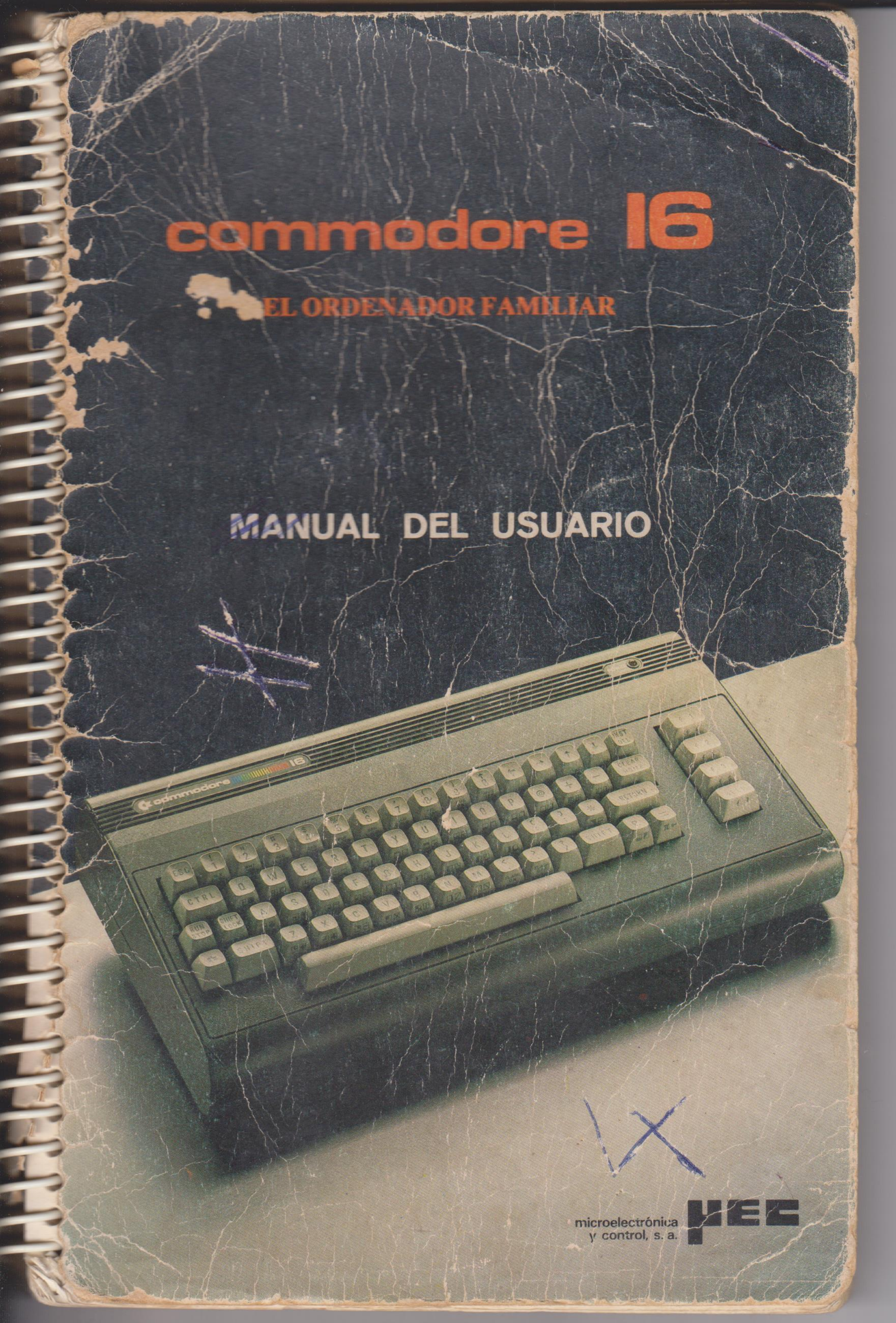
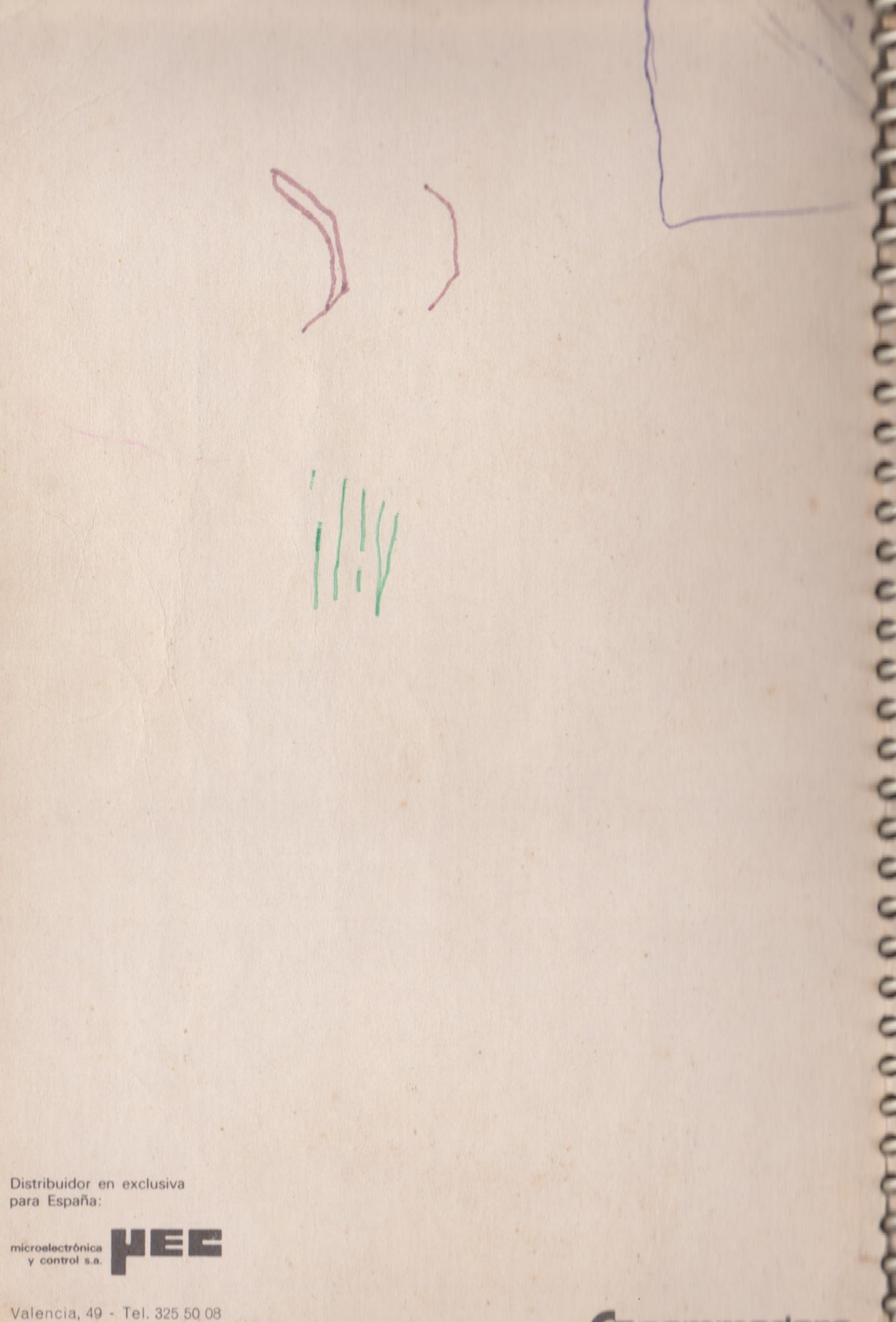
commodore 16

EL ORDENADOR FAMILIAR

MANUAL DEL USUARIO



microelectrónica y control, s. a. MEC



Distribuidor en exclusiva
para España:

microelectrónica
y control s.a.
MEC

Valencia, 49 - Tel. 325 50 08
Telex: 50916 myc b - 08015 BARCELONA

Princesa, 47-3.° G - Tel. 248 95 70 - 28008 MADRID

commodore

COMMODORE 16
MANUAL DEL USUARIO

Copyright © 1984 by Commodore Business Machines, Inc.
Todos los derechos reservados.

Ninguna parte de esta publicación puede ser reproducida, grabada ni transmitida en forma alguna y por no importa que medio electrónico, mecánico, de fotocopia u otro tipo sin previo consentimiento por escrito de Commodore Internacional.

INDICE

CAPITULO 1
PUESTA EN MARCHA

- Desembalado de su COMMODORE 16
- Toma de contacto con los interruptores y conexiones
- Puesta en marcha de su COMMODORE 16
- Tabla de Problemas
- Periféricos

DESEMBALADO DE SU COMMODORE 16

Ahora que ha abierto Vd. la caja que contiene su nuevo COMMODORE 16 y ha encontrado este manual, lo primero que debería hacer es asegurarse de que tiene todos los artículos contenidos en esta lista. Debería Vd. tener:

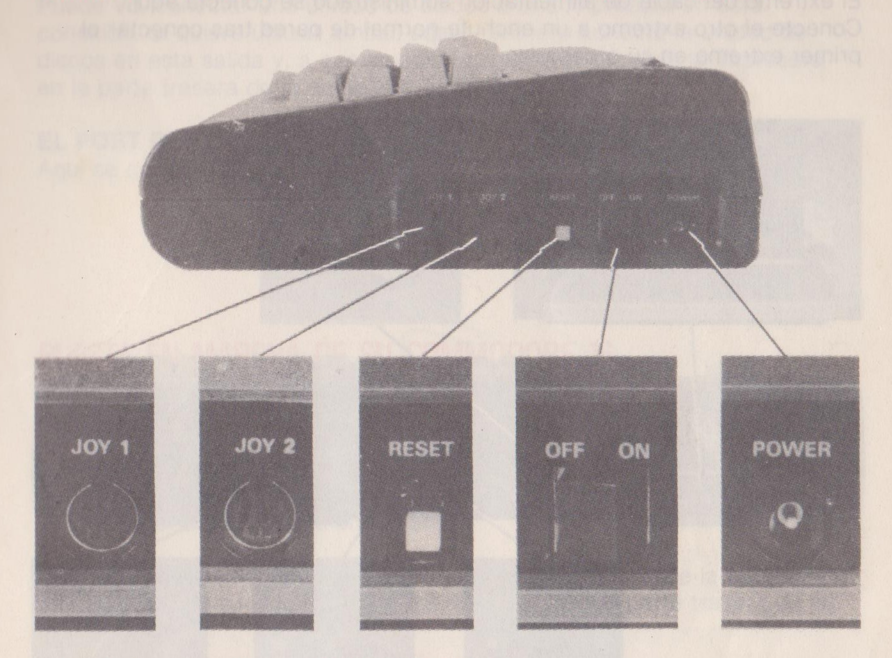


1. Su COMMODORE 16
2. El cable de conexión a la corriente
3. El cable de conexión al televisor
4. El manual del usuario. (Probablemente lo habrá encontrado, ya que está ahora leyéndolo)
5. Otros escritos varios:
   Tarjeta de garantía

Si no ha encontrado Vd. todos estos artículos en la caja, solicítelos inmediatamente a su distribuidor.

Antes de conectar cualquier cosa, debe echar un vistazo a estas fotografías de su ordenador. Puede familiarizarse con todas las conexiones e interruptores, así podrá poner en marcha su sistema rápida y fácilmente.

TOMA DE CONTACTO CON LOS INTERRUPTORES Y CLAVIJAS

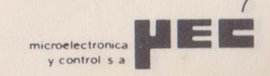


Lateral derecho de su COMMODORE 16

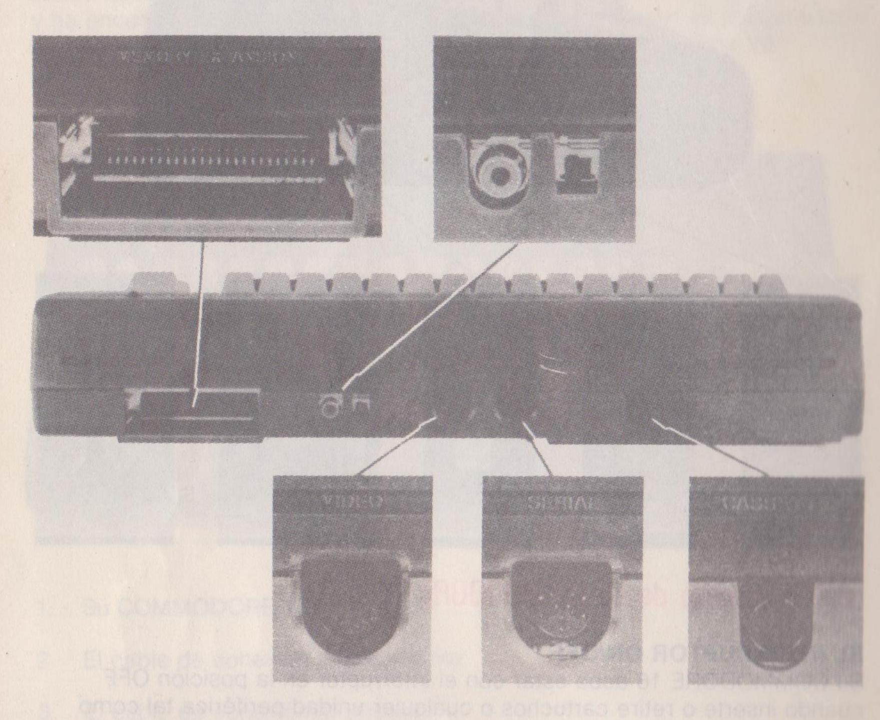
EL INTERRUPTOR ON/OFF
Su COMMODORE 16 debe estar con el interruptor en la posición OFF cuando inserte o retire cartuchos o cualquier unidad periférica tal como una impresora o la unidad de cintas Datacassette. Hay una luz roja colocada en la parte superior derecha, sobre el teclado, que está encendida cuando el ordenador está conectado con el interruptor en la posición ON.

LAS CONEXIONES PARA JOYSTICK (PALANCA DE JUEGOS)
Puede Vd. conectar joysticks en estas clavijas, rotuladas JOY#1 y JOY#2. El COMMODORE 16 utiliza joysticks especialmente diseñados disponibles en su distribuidor COMMODORE.

EL BOTON DE RESET (BORRADO)
Pulse el botón de reset cuando desee "volver a empezar" igual que si hubiera Vd. conectado nuevamente su ordenador. El botón de reset limpia la pantalla y borra cualquier programa BASIC que hubiera tecleado previamente. Hay otros modos de inicializar su ordenador sin borrar los programas. Esto se detallará en el capítulo 4.



LA CLAVIJA DE ALIMENTACION
El extremo del cable de alimentación suministrado se conecta aquí. Conecte el otro extremo a un enchufe normal de pared tras conectar el primer extremo en su ordenador.



PARTE TRASERA DE SU ORDENADOR

EL PORT DE EXPANSION DE MEMORIA
Los cartuchos de software para su COMMODORE 16 deben conectarse aquí. Antes de insertarlos o retirarlos, asegúrese de que el ordenador esta en OFF.

El JACK RF
Aquí es donde conectará Vd. un extremo del cable del televisor (el cable negro y delgado). Un extremo del cable irá en este jack y el otro extremo en la conexión de antena de su televisor.

LA CONEXION DE VIDEO
Aquí es donde colocará Vd. el cable que conecte un monitor a su COMMODORE 16. Si utiliza Vd. un aparato de televisión, no necesitará usar esta conexión.



EL BUS SERIE
Puede Vd. conectar una unidad de discos o una impresora en esta conexión. Si quiere utilizar ambas cosas, primero conecte la unidad de discos en esta salida y, a continuación conecte el cable de la impresora en la parte trasera de la unidad de discos.

EL PORT DEL CASSETTE
Aquí se conecta el grabador de cintas COMMODORE 1531 Datassette.

PUESTA EN MARCHA DE SU COMMODORE 16

Conectar su COMMODORE 16 es simple. Solo necesita hacer dos cosas:

1 Conectar el cable de alimentación desde el lado derecho de su ordenador a un enchufe en la pared.

2 Conectar el cable del televisor (el negro y delgado) desde la conexión de antena de su aparato a la conexión de RF en la parte trasera de su ordenador.

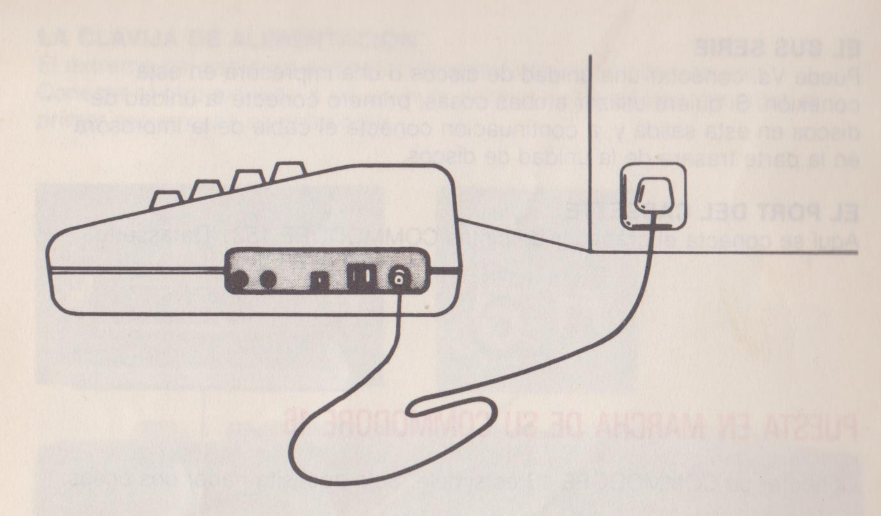
Asegúrese de que tiene suficientes enchufes para conectarlo todo cuando decida Vd. Poner en marcha su ordenador. Puede necesitar una bateria de enchufes, especialmente si quiere utilizar también una impresora o una unidad de discos. Recuerde que todo (ordenador, televisor,etc), debe estar apagado hasta que esté Vd. completamente listo para empezar.

1: Conexión del cable de alimentación

- Primero conecte el lado redondo del cable de alimentación de la conexión lateral derecho de su ordenador.

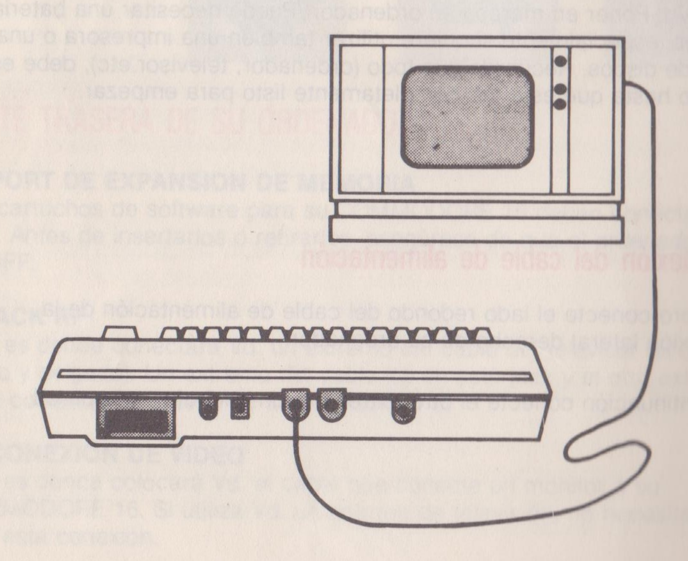
- A continuación conecte el otro extremo a un enchufe en la pared.





2: Conexión del cable del televisor

- Conecte un extremo del cable de TV (solo un extremo) a la entrada de antena de su televisor.
- Conecte el otro extremo del cable al jack RF en la parte trasera de su ordenador.



Conexión de su Commodore 16 a un monitor

Si conecta Vd. su ordenador a un monitor en vez de hacerlo a un aparato de televisión, siga las instrucciones indicadas en el manual que se incluye con el monitor. Conectar un monitor tal como el Monitor en Color Commodore 1702 es sencillo. Solamente necesita conectar un cable que va directamente de la entrada del monitor a la salida trasera de su ordenador.

Finalmente

Ahora es el momento de conectar su ordenador. (Si ha prestado Vd. atención debería saber donde se encuentra el interruptor de puesta en marcha.)

El siguiente paso consiste en sintonizar su televisor. El COMMODRE 16, a través de su conexión de Alta Frecuencia, se comporta como una emisora de televisión que ''emite'' por el canal 36 (aproximadamente) de la banda de UHF. En todo caso, es conveniente que consulte el manual de instrucciones de su televisor, no obstante le adjuntamos unas breves instrucciones que suelen servir para la mayoría de los televisores que existen actualmente. Para sintonizar su televisor escoja un botón de sintonía que no esté programado para las emisoras que Vd. utiliza normalmente y —con el COMMODORE 16 conectado a la toma de antena de su televisor— mueva la sintonía en la banda de UHF hasta encontrar el texto que sigue a este párrafo, en este momento tiene Vd. sintonizado su televisor. En algunos televisores antiguos la toma de antena es doble, una para VHF y otra para UHF, en este caso debe proveerse de un conversor-separador de 75 a 300 ohmios —que encontrará en la tienda de electrónica más cercana— e intercalarlo entre el cable del COMMODORE 16 y su televisor. Si todo ha ido bien se encenderá el indicador rojo en el ordenador y en la pantalla del televisor aparecerá el mensaje:

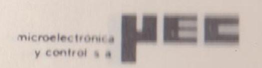
```
COMMODORE BASIC V3.5 12277 BYTES FREE
READY
■
```

El cursor parpadeante bajo el mensaje de READY le indica que su COMMODORE 16 está esperando que Vd. empiece a teclear. El color del fondo es blanco, mientras que las letras se imprimen en negro con un borde púrpura alrededor de la pantalla.

Si ha tenido alguna dificultad la tabla de problemas puede serle de ayuda.

TABLA DE PROBLEMAS

Sintonía	Causa	Remedio
Luz indicadora apagada	Ordenador desconectado	Asegúrese de que el interruptor está en ON
	Cable de alimentación mal conectado	Compruebe si el cable está bien conectado
	Cable de alimentación desenchufado	Compruebe la conexión con el enchufe en la pared
	Fusible fundido en el ordenador	Lleve su aparato a un distribuidor autorizado para que reemplace el fusible
No hay imagen	Cable del televisor desconectado	Comrpruebe la conexión con el televisor
	Televisor apagado	Póngalo en marcha
Signos aleatorios en la pantalla con un cartucho insertado	Cartucho mal colocado	Vuelva a colocar el cartucho tras apagar el ordenador
Imágen sin color	Televisor fuera de canal	Sintonice el canal correctamente
	Conexión inadecuada al televisor	Compruebe las conexiones
	Colores demasiado débiles	Ajuste los colores
Imágen bien, pero sin sonido	Volumen demasiado bajo en el televisor	Ajuste el volumen en el televisor
	Televisor fuera de canal	Sintonice el canal correctamente



IMPORTANTE: Algunos televisores no pueden representar la pantalla del COMMODORE 16 en su totalidad, cortando las columnas finales de ambos lados. Le recomendamos que, en ese caso, utilice otro televisor o bien un monitor como los Commodore 1702, 1802 o 1803.

Si esto no es posible, puede Vd. solventar el problema pulsando la tecla ESC y, a continuación, la tecla ''R''. De este modo la representación en la pantalla se reduce a 38 columnas, por lo cual ya cabrá toda la representación en la pantalla. Deberá repetir esta operación cada vez que conecte o restaure el ordenador.

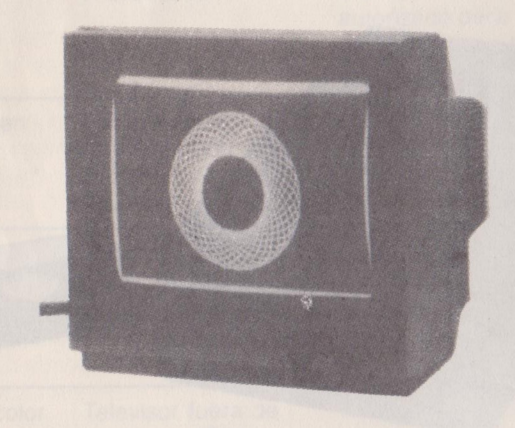
PERIFERICOS

Los periféricos son los accesorios que puede Vd. utilizar junto a su COMMODORE 16 incrementando sus posibilidades. Estos accesorios están disponibles en su distribuidor Commodore y le permiten utilizar su ordenador al máximo.



Los periféricos le dan la capacidad de transferir y almacenar datos, imprimir lo que aparece en la pantalla (en blanco y negro), utilizar programas preparados y almacenados en cassette o en diskette floppy y darle una imágen clara de los resultados de su ordenador.

Para almacenar o volver a ejecutar programas, necesitará una unidad que almacene datos. Estos pueden ser registrados y recuperados a través de cintas de cassette y diskette. Para utilizar programas en cinta (y para grabar sus propios programas en un cassette), necesitará la grabadora Commodore 1531 DATASSETTE. Para diskette, hay varias UNIDADES DE DISCO disponibles. Las unidades de disco son rápidas y de un uso muy eficiente. Las unidades compatibles con su COMMODORE 16 son los modelos Commodore 1541 y 1551.



Su aparato de televisión quizá no le dé una imagen todo lo clara que Vd. desearía. Los monitores en color Commodore están especialmente diseñados para darle la imagen más brillante y más clara de las salidas de su COMMODORE 16. Hay varios modelos disponibles, incluyendo el Commodore 1702, 1802 y 1803.



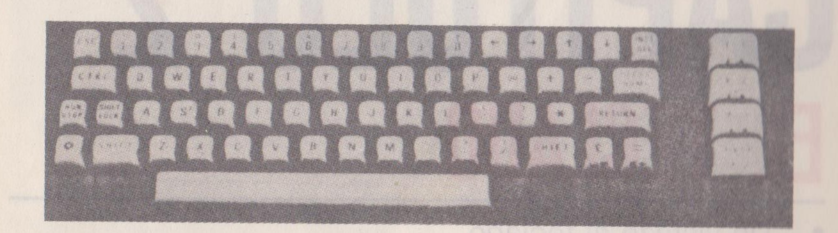
Cuando utilice un programa procesador de textos o un paquete de gráficos en su COMMODORE 16, una impresora le representará en papel lo que hay en la pantalla. Hay varios modelos de impresoras COMMODORE capaces de trabajar con su ordenador. Entre ellas se incluyen la MPS-801, MPS-802, MPS-803 (con alimentación por tractor opcional), y DPS-1101 (alta calidad de impresión).

CAPITULO 2

EL TECLADO

- Un vistazo al teclado
- Teclas especiales
- Teclas de gráficos
- Teclas de función
- La tecla HELP

UN VISTAZO AL TECLADO



La mayor parte del teclado de su COMMODORE 16 es idéntico a las teclas de una máquina de escribir, pero cada tecla puede hacer más cosas que una de estas; En esta sección, aprenderá a usar las teclas especiales como la tecla C= y las cuatro teclas del cursor. Esta sección le mostrará las funciones extra de cada tecla, incluyendo como utilizar los símbolos gráficos dibujados en la parte frontal de muchas de ellas. Con cada explicación acerca de las diferentes teclas del teclado de su ordenador, deberá Vd. buscarlas y particar con ellas.

Utilización del teclado como una máquina de escribir

Cuando Vd. empieza a escribir letras en su ordenador, aparecen en la pantalla como mayúsculas. Las letras y los números aparecen en la pantalla exactamente igual que en las teclas. También hay algunas otras (+, −, =, @, *, y £) que pueden teclearse conjuntamente con la tecla SHIFT. Si quiere Vd. hacer un escrito "regular", puede teclar en mayúsculas y minúsculas (como lo haría en una máquina de escribir) pulsando las teclas **SHIFT** y C= al mismo tiempo. Tras hacerlo, todas las letras que se pulsen aisladamente, se imprimirán en minúsculas. Cuando pulse la tecla **SHIFT** conjuntamente con una letra, tendrá una mayúscula. Los números y las teclas de puntuación trabajan como si estuvieran en el modo habitual (no mecanográfico). Para salir de este modo mecanográfico, pulse inmediatamente las dos teclas (**SHIFT** y C=) de nuevo conjuntamente.

KEY	+	KEY =		
SHIFT	+	C=	=	ENTRAR O SALIR DEL MODO MECANOGRAFICO
A			=	a
SHIFT	+	A	=	A
C=	+	A	=	┌

Teclas especiales

Algunas teclas de su COMMODORE 16 actúan de modo bastante diferente al de cualquier tecla de una máquina de escribir. Estas teclas actúan activando a otras teclas para que hagan cosas que ordinariamente no harían, o para ejecutar funciones relacionadas con el ordenador. El teclado de su COMMODORE 16 contiene también símbolos especiales que no se encuentran en las demás máquinas de escribir e, incluso, en muchos ordenadores. Estos signos especiales incluyen el signo de la libra (£) la letra Pi (π), los signos mayor que y menor que (< >), corchetes ([ ]) y las flechas (↑). Estas teclas de signos especiales son usadas a menudo al escribir programas en su COMMODORE 16.

RETURN

Deberá Vd. pulsar la tecla **RETURN** al final de cualquier linea de instrucciones que introduzca a través del teclado de su COMMODORE 16. Puede pensar que esta tecla es una tecla **ENTER** ya que **RETURN** actúa entrando información e instrucciones a su ordenador.

SHIFT

Ya ha visto Vd. un ejemplo del efecto de la tecla **SHIFT** utilizando el teclado como una máquina de escribir. Esto es lo típico de la tecla **SHIFT.** Se utiliza siempre para modificar lo que otras teclas imprimen en la pantalla, pero no puede hacer nada por si misma. La tecla **SHIFT** le permite teclear mayúsculas, gráficos, signos de puntuación y algunas otras cosas, con ayuda de otras teclas. Verá más funciones de la tecla **SHIFT** a lo largo de esta sección, como la obtención de símbolos gráficos.

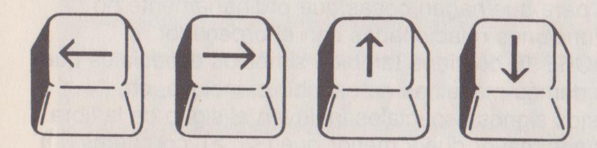
La tecla **SHIFT LOCK** es lo mismo que la tecla **SHIFT,** a excepción de que queda bloqueada, por lo que no necesita mantenerla pulsada. Cuando **SHIFT LOCK** está activada, cada carácter que teclee es representado como si pulsara la tecla **SHIFT.** para liberar el bloqueo, simplemente pulse de nuevo la tecla **SHIFT LOCK** y cualquier cosa que teclee habrá vuelto a la normalidad.

RUN-STOP

Pulse esta tecla para cortar la ejecución de un programa y parar todo lo que su COMMODORE 16 está haciendo. Cuando su ordenador está ejecutando un programa, la pulsación de esta tecla devuelve el control al teclado.

Cuando pulse **SHIFT** y **RUN-STOP** simultáneamente, el COMMODORE 16 cargará y ejecutará el primer programa del disco que se encuentre en la unidad de discos.

Las teclas del cursor



El cursor, el cuadrado parpadeante que le indica la posición en la que se encuentra en la pantalla del ordenador, puede ser movido a través de ella rápida y fácilmente utilizando las TECLAS DEL CURSOR. Hay cuatro teclas del cursor separadas con una flecha cada una apuntando en la dirección en la que mueven el cursor; arriba, abajo, izquierda y derecha. Puede Vd. utilizar estas teclas para mover el cursor a cualquier punto de la pantalla sin afectar para nada a los caracteres que haya en ella. Al igual que las demás teclas, las teclas del cursor tienen repetición automática. Esto significa que si Vd. mantiene pulsada la tecla, el cursor se mueve continuamente en la dirección indicada en dicha tecla.

INST/DEL

Mediante esta tecla puede Vd. INSERTAR Y BORRAR letras y números en la linea en que se encuentre. Cuando pulsa Vd. esta tecla aisladamente, el carácter inmediatamente a la izquierda de la posición del cursor desaparece y el cursor retrocede a la posición que ocupaba dicho carácter. Puede Vd. utilizar las teclas del cursor para volver a la mitad de una linea y, una vez allí, utilizar **DEL** para borrar una letra. Cuando hace esto, la letra de la izquierda es borrada y el resto de las letras de la linea se mueven un espacio a la izquierda para cerrar el hueco.

Para abrir un espacio en el que insertar letras y números pulse esta tecla conjuntamente con **SHIFT.** El espacio se abre a la derecha del cursor. El cursor mismo no se mueve. Cuando inserta Vd. un espacio en medio de una linea de letras el resto de la linea se mueve a la derecha.

La tecla **INST/DEL** ahorra gran cantidad de tiempo cuando quiere Vd. corregir o cambiar lo que ha tecleado.

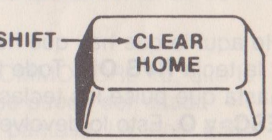
TECLA	+		=	EFECTO
INST/DEL			=	CORREGIR (BORRAR CARACTERES TECLEADOS)
SHIFT	+	**INST/DEL**	=	INSERTAR (AÑADIR ESPACIOS)

SHIFT → INST DEL

CLEAR/HOME

Esta tecla sirve para dos funciones primarias: HOME y CLEAR. Cuando Vd. pulsa esta tecla aisladamente, el cursor inmediatamente va a la esquina superior izquierda de la pantalla (que se conoce como la posición de HOME). El resto de la pantalla permanece igual. Si pulsa Vd. la tecla **SHIFT** y **CLEAR/HOME** no solamente se mueve el cursor a la posición HOME sinó que toda la pantalla es borrada (o limpiada). Lo único que permanece en la pantalla es el cursor parpadeante en la esquina superior izquierda.

TECLA	+		=	EFECTO
CLEAR/HOME			=	POSICION HOME
CLEAR/HOME	+	**SHIFT**	=	BORRA PANTALLA



CTRL (Control)

La letra **CTRL** es como la tecla **SHIFT** en el sentido de que trabaja conjutamente con la tecla. Debe mantenerla pulsada mientras pulsa otras teclas. **CTRL** se utiliza en tres ocasiones:

1) Tal como se explica en la sección de TECLAS DE COLOR, pulsando **CTRL** y una tecla de color cambiará el color del texto impreso en la pantalla.
2) Puede Vd. detener un programa que se esté imprimiendo o listando en la pantalla pulsando **CTRL** y la tecla S.
   (Para volver a reiniciar el programa, pulse cualquier tecla excepto **RUN/STOP.)**
3) **CTRL** se utiliza con **REVERSE ON/OFF** y **FLASH ON/OFF.** Esto se explicará más adelante en esta misma sección.

Además, algunos programas de los que adquiera Vd. hacen uso de la tecla **CTRL** para sus propias funciones especiales.

C= (TECLA COMMODORE)

La tecla Commodore es muy similar a **CTRL** y puede ser utilizada para realizar cuatro funciones:

1) Cuando se utiliza con la tecla **SHIFT,** la tecla **C=** le permite trabajar en el modo mecanográfico, en el que puede Vd. utilizar letras mayúsculas y minúsculas.
2) La tecla **C=** le permite siempre teclear el símbolo gráfico representado a la izquierda de la parte frontal de cada tecla. Simplemente pulse la tecla **C=** y la tecla de gráfico que desee.
3) La tecla **CCC** se utiliza como la tecla **CTRL** para cambiar el color de lo que teclee en la pantalla cuando es pulsada conjuntamente con una TECLA DE COLOR.
4) Cuando quiera ralentizar un listado (un programa que aparezca línea a línea en la pantalla demasiado rápidamente para seguirlo), pulse la tecla **C=**. La velocidad se reducirá considerablemente. Cuando suelte Vd. la tecla, volverá a su velocidad normal, (la tecla **C=** puede hacer algo por sí misma).

RVS ON RVS OFF (Inverso On/Off)

El COMMODORE 16 le permite imprimir las letras y números en imagen inversa. En otras palabras, si el cursor es negro y el fondo es amarillo, lo que teclee aparecerá en letras amarillas sobre fondo negro.

He aquí lo que hay que hacer para invertir imágenes: pulse la tecla **CTRL** y la tecla **RVS ON.** Todo lo que teclee aparecerá ahora en modo inverso, hasta que pulse las teclas **CTRL** y **RVS OFF,** la tecla **RETURN** o la tecla **ESC** y **O.** Esto lo devolverá al modo normal (caracteres no inversos).

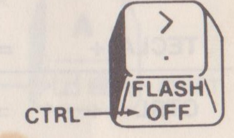
TECLA	+		=	EFECTO
CTRL	+	RVS ON	=	ACTIVADA IMPRESION INVERSA
CTRL	+	RVS OFF	=	IMPRESION NORMAL



FLASH ON FLASH OFF

Vd. puede hacer que los caracteres en la pantalla parpadeen continuamente al igual que lo hace el cursor. Pulse **CTRL** y la tecla **FLASH ON** para conseguir que todo lo que teclee parpadee. Tecleando **CTRL** y **FLASH OFF,** o **RETURN,** o **ESC** y **O** volverá de nuevo al modo normal (no parpadeante).

TECLA	+	=	EFECTO
CTRL	+ FLASH ON	=	CARACTERES PARPADEANTES
CTRL	+ FLASH OFF	=	REPRESENTACION NORMAL

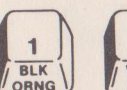


ESC (Escape)

La tecla **ESC** le permite efectuar algunas funciones especiales del editor de pantalla, incluyendo funciones para activar y manipular ventanas de pantalla.

Las ventanas son una propiedad especial de su COMMODORE 16, que le permite mantener aparte un area de la pantalla que puede ser usada como espacio de trabajo sin afectar al resto de la pantalla. La tecla **ESC** puede efectuar varias funciones de edición de ventanas, así como otros usos habituales tales como inserción, borrado y deslizamiento (SCROLL). Todas las funciones de la tecla **ESC** serán listadas y explicadas cuando veamos las ventanas de pantalla en el capítulo 4.

Teclas de color



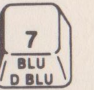












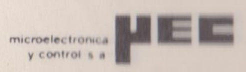


Las teclas de color guardan una notable semejanza con las teclas numéricas del 1 al 8. Cuando se pulsa una de ellas con las teclas **CTRL** o **C=**, cambia el color de lo que teclee Vd. en la pantalla. Su ordenador empieza a trabajar con un fondo blanco y borde azul, con el cursor

parpadeando en negro. Cuando teclea Vd. algo, el carácter aparece en negro. Para cambiar el cursor (y todo lo que teclee) a otro color diferente, utilice las teclas de color. En la parte frontal de las teclas numeradas de 1 a 8 hay escritos dos colores. Pulse **CTRL** junto con una tecla numérica para obtener el color superior, o la tecla **C=**, junto con el número para obtener el inferior. Puede Vd. cambiar solamente el color de los caracteres utilizando las teclas de color. Para cambiar los colores del fondo o del borde, debe usar un comando BASIC. (Se explicará más adelante.)

> **NOTA:** Cuando cambie los colores, tenga cuidado de no pulsar la tecla del número (color) incluso un instante antes de pulsar **CTRL** o **CCC.** Si lo hace, todo lo que hará es imprimir un número en la pantalla en vez de cambiar de color.

TECLA +		= EFECTO	TECLA +		= EFECTO
CTRL	+ 1	= NEGRO	**C=**	+ 1	= NARANJA
CTRL	+ 2	= BLANCO	**C=**	+ 2	= MARRON
CTRL	+ 3	= ROJO	**C=**	+ 3	= VERDE AMARILLO
CTRL	+ 4	= CIAN	**C=**	+ 4	= ROSA
CTRL	+ 5	= PURPURA	**C=**	+ 5	= VERDE AZUL
CTRL	+ 6	= VERDE	**C=**	+ 6	= AZUL CLARO
CTRL	+ 7	= AZUL	**C=**	+ 7	= AZUL OSCURO
CTRL	+ 8	= AMARILLO	**C=**	+ 8	= VERDE CLARO



Teclas de gráficos

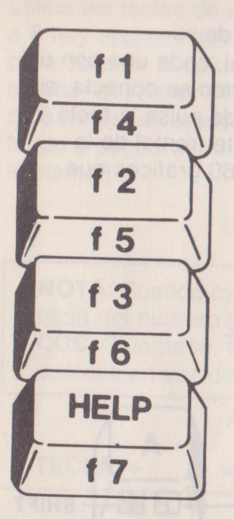
Cada tecla de letra (y algunas otras teclas de símbolos) de su COMMODORE 16 tiene dos recuadros en la parte frontal, cada uno con un símbolo diferente. Estas son las teclas de gráficos. Cuando se conecta, su ordenador imprime primeramente en mayúsculas. Cuando pulsa la tecla **SHIFT** o **C=**, escribe los gráficos mostrados en la parte frontal de la tecla. Puede Vd. imprimir el juego completo de más de 60 gráficos que puede ver la parte frontal de muchas de las teclas.

TECLA	+	=	EFECT
SHIFT	+ A	=	♠ GRAFICO A LA DERECHA DE LA TECLA
CCC	+ A	=	┌ GRAFICO A LA IZQUIERDA DE LA TECLA

Puede Vd. hacer dibujos, tablas y diseños imprimiendo gráficos uno junto a otro como si fueran un puzzle. Puede hacer todavía más interesantes estos gráficos utilizando las teclas de color para "dibujarlos" en diferentes colores. Intente imprimir algo con las teclas de gráficos para ver como trabajan. El Capítulo 6 explica más cosas sobre gráficos.

Cuando está Vd. en modo mecanográfico, puede usar únicamente los símbolos gráficos de la izquierda (pulsando **C=** con la tecla apropiada). Los gráficos del lado izquierdo son ideales para formar tablas, gráficos y formularios administrativos.

Teclas de función



Las cuatro teclas a la derecha de su teclado (separadas del resto de las teclas son teclas de funciones especiales que le permiten ahorrar tiempo al realizar tareas repetitivas que podrá realizar pulsando una única tecla. En la parte superior de cada tecla está escrito f1, f2, f3, y HELP. Puede Vd. obtener estas funciones pulsando estas teclas por separado. La parte frontal de estas teclas esta escrito f4, f5, f6 y f7. Pulsando **SHIFT** y f1, f2, f3 y HELP, respectivamente, obtendrá estas funciones.

He aquí lo que hace cada tecla:

TECLA F1 entra uno de los modos de GRAPHIC cuando introduce Vd. el número del área de gráficos y pulsa **RETURN.** El comando GRAPHIC es necesario para dar comandos de gráficos tales como CIRCLE o PAINT. Más explicaciones sobre GRAPHIC las encontrará en el Capítulo 6.

TECLA F2 imprime DLOAD'' en la pantalla. Todo lo que debe hacer es introducir un nombre de programa para cargar un programa desde disco y pulsar **RETURN** en vez de teclear DLOAD'' Vd. mismo.

TECLA F3 lista el directorio de los ficheros del disco que hay en la unidad de disco.

TECLA F4 borra la pantalla usando el comando SCNCLR.

TECLA F5 imprime DSAVE'' en la pantalla. Todo lo que debe hacer es introducir el nombre de un programa para registrarlo en un disco y pulsar **RETURN** en vez de tener que teclear DSAVE'' manualmente.

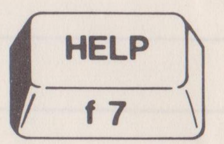
TECLA F6 ejecuta el programa en memoria.

TECLA F7 saca un listado del programa en memoria.

TECLA F8 HELP destaca los errores en el listado de un programa imprimiéndolos en caracteres parpadeantes.

Puede Vd. redefinir cualquiera de estas teclas para que desarrollen funciones que satisfagan sus necesidades. El redefinirlas es fácil, utilizando el comando KEY. Puede Vd. redefinirlas a través de sus programas BASIC o cambiarlas en cualquier momento en modo directo. Una ocasión en la que puede Vd. desear redefinir una función es cuando utiliza frecuentemente un mismo comando y quiere ahorrar tiempo evitando teclearlo repetidamente. Las nuevas definiciones son borradas cuando se desconecta el ordenador. Puede Vd. redefinir tantas teclas como desee.

La tecla Help



Cuando comete un error en un programa, su ordenador presentará un mensaje de error diciéndole que algo anda mal. Estos mensajes están explicados más adelante en los apéndices de este manual.

Puede Vd. obtener una mayor ayuda en caso de errores utilizando la tecla **HELP.** Tras un mensaje de error, pulse la tecla para localizar el error exactamente. Cuando pulsa HELP, la línea con el error es representada en la pantalla con el error parpadeando. Por ejemplo:

?SYNTAX ERROR IN LINE 10	Su ordenador presenta un mensaje de error
HELP	Vd. pulsa HELP
10 PRONT "COMMODORE COMPUTERS"	El error es presentado en modo parpadeante



CAPITULO 3

SOFTWARE

- Introducción
- Cartuchos
- Cassettes
- Diskettes

INTRODUCCION

Un ordenador sin software es como un vaso sin naranjada. Bueno, quizá no sea así, pero el software la utilidad de su ordenador y la alegría y diversión que le proporciona. El software son todos los programas que puede Vd. introducir y ejecutar con un ordenador. El hardware (en este caso, su COMMODORE 16) puede utilizar el software del muchas maneras: conectando cartuchos, cintas y diskettes pregrabados. La gama de software disponible en su COMMODORE 16 está aumentando rápidamente. Su distribuidor puede mantenerle informando de los nuevos productos y de las novedades disponibles en la actualidad.

Su COMMODORE 16 puede utilizar software en forma de CARTUCHOS, CINTAS CASSETTE y DISKETTES, disponibles en su distribuidor Commodore. Todo lo que debe hacer es cargarlos en su COMMODORE 16. También puede crear sus propios programas y almacenarlos en cintas de cassette o en discos floppy.

CARTUCHOS

Commodore produce un amplio surtido de software en cartuchos para el COMMODORE 16. Hay gran variedad de programas personales, educativos y comerciales, así como excitantes programas de juegos a su disposición. No necesita Vd. ningún equipo adicional para utilizar el software en cartuchos. Lo único que debe hacer es conectar el cartucho en la parte posterior de su ordenador y conectar éste. He aquí los pasos que debe seguir para utilizar un cartucho:

Carga de cartuchos

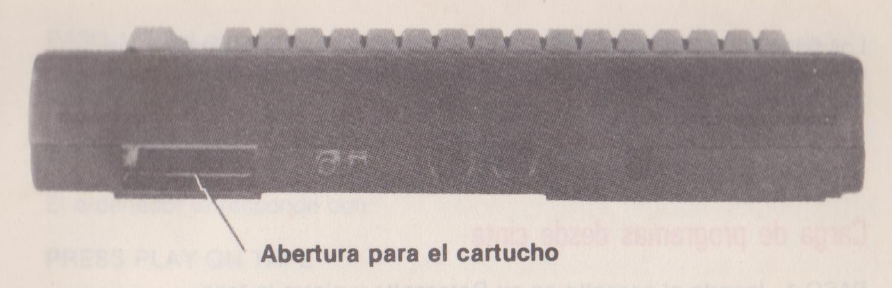
PASO 1 Apague su COMMODORE 16.

> **IMPORTANTE:** DEBE VD. APAGAR SU ORDENADOR ANTES DE INSERTAR O RETIRAR EL CARTUCHO. SI NO LO HACE ASI, PUEDE AVERIAR EL CARTUCHO Y EL ORDENADOR.

PASO 2 Sostenga el cartucho con la etiqueta hacia arriba e insértelo firmemente en el hueco correspondiente (rotulado ''memory'' expansión'') en la parte trasera del ordenador.

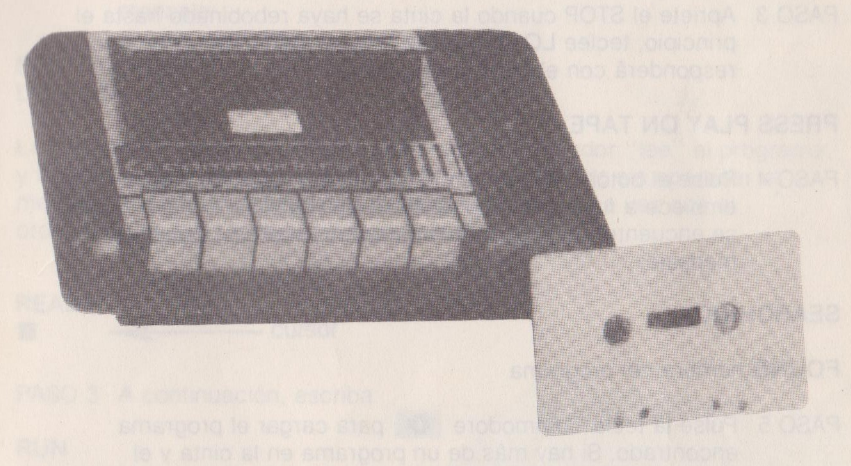
PASO 3 Conecte su COMMODORE 16.

PASO 4 Empiece el juego o el programa de acuerdo con las instrucciones que lleve adjuntas el propio cartucho.



Abertura para el cartucho

CASSETTES



Hay una gran variedad de software para el COMMODORE 16 disponible en forma de cassette. Estas cintas son similares a las cassettes de música que utiliza Vd. en su grabadora o en su cadena HI-FI. El software en cinta trabaja del mismo modo que en los cartuchos pero debe Vd. tener una pieza adicional en su equipo (un periférico) para cargar los programas desde el cassette a su ordenador. Para ello necesita un grabador de cintas Datassette, que podrá encontrar en su distribuidor Commodore.

También puede utilizar las cintas y el Datassette para almacenar programas escritos por Vd. mismo. La siguiente sección le muestra como guardar programas en cinta.

Los pasos a seguir para cargar un programa en cinta son siempre los mismos, ya sea una cinta pregrabada o una con programas hechos por Vd. mismo.

Carga de programas desde cinta

PASO 1 Inserte el cassette en su Datassette y cierre la tapa.
PASO 2 Rebobine la cinta hasta el comienzo pulsando el botón REWIND en el Datassette.
PASO 3 Apriete el STOP cuando la cinta se haya rebobinado hasta el principio, teclee **LOAD** y pulse la tecla **RETURN.** El ordenador responderá con el mensaje siguiente:

PRESS PLAY ON TAPE

PASO 4 Pulse el botón de PLAY en el Datassette. Cuando el Datassette empiece a funcionar la pantalla se quedará sin imágen. Cuando se encuentre un programa, aparecerá en ella el siguiente mensaje:

SEARCHING

FOUND nombre del programa

PASO 5 Pulse la tecla Commodore C= para cargar el programa encontrado. Si hay más de un programa en la cinta y el encontrado por el COMMODORE 16 no es el que Vd. desea, no haga nada. Su ordenador continuará buscando tras un breve intervalo.

Cuando el programa está cargado aparecerá en la pantalla la palabra READY. Si quiere detener la carga antes de que se complete, pulse **RUN/STOP** y, a continuación, apriete el botón de STOP en el Datassette. Tras haber cargado el programa, teclee **RUN** para ejecutarlo. Puede también listarlo o cambiarlo si se trata de un programa en BASIC.

Carga de un programa específico

Para cargar un programa específico desde la cinta, utilice el comando **LOAD** "nombre del programa". Esta instrucción es la misma que **LOAD** sin ningún nombre, con unas leves diferencias.

PASO 1 Si el programa que desea se llama "BASES", debe Vd. escribir:

LOAD "BASES"

y pulsar **RETURN**

El ordenador le responde con:

PRESS PLAY ON TAPE

PASO 2 Apriete el botón de PLAY en el Datassette para que el ordenador empiece a buscar. Tras la búsqueda, deberá aparecer el mensaje:

FOUND BASES
LOADING

La pantalla se queda sin imágen mientras el ordenador "lee" el programa y lo almacena en la memoria. Si la cinta sigue girando sin aparecer el mensaje FOUND, rebobínela e inténtelo de nuevo. Una vez que el ordenador ha leído todo el programa, le dirá;

READY
■ ← cursor

PASO 3 A continuación, escriba:

RUN

y pulse **RETURN.** En ese momento, el COMMODORE 16 ejecuta el programa "BASES".

Grabación de programas en cinta

Cuando escriba sus propios programas y desee guardarlos para un uso posterior o para modificarlos más adelante, puede SALVARLOS. Cuando Vd. salva un programa, en realidad lo está grabando en algún medio (cinta o disco) que le permite volver a recogerlo y ejecutarlo de nuevo, hacer cambios, etc. Cuando desee salvar un programa en una cinta de cassette siga estos pasos:

PASO 1 Escriba:

SAVE "nombre del programa"

El nombre del programa puede ser cualquiera, pero no puede ser mayor de 16 caracteres.

PASO 2 Pulse la tecla **RETURN.** El ordenador le dará el siguiente mensaje:

PRESS PLAY AND RECORD ON TAPE

PASO 3 Apriete los botones RECORD y PLAY en el Datassette. La pantalla se queda sin imágen. Cuando su programa esté salvado, aparecerá en la pantalla la palabra READY.

Ejemplos del comando SAVE para cinta:

SAVE "DIA" SAVE "VD MISMO"	Este es el nombre específico del programa a salvar

> **NOTA:** Cuando salve un programa en cinta, fíjese siempre donde está colocada la cinta.

DISKETTES



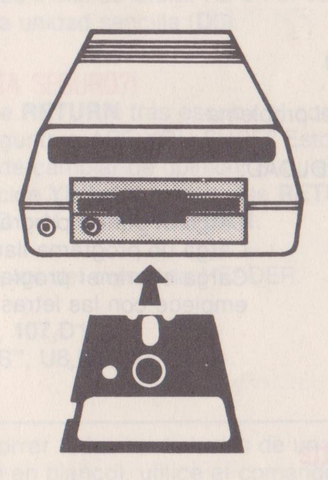
Los discos son rápidos y fáciles de usar. Asegúrese de manejar con cuidado los discos y la unidad de discos. Los discos pueden ser llamados diskettes, discos floppy o floppys indistintamente. Las tres cosas son lo

mismo. A diferencia de los cassettes, Vd. solo debe colocar el disco dentro de la unidad y teclear los comandos para SALVAR o CARGAR programas. No hay botones que apretar. Hay un par de lucecitas en la parte frontal de la unidad. La luz verde es el indicador de conexión que le indica si la unidad está encendida o apagada. La luz roja le indica dos cosas. Durante el uso normal de la unidad, cuando está siendo salvado o cargado un programa, se enciende mientras el disco está girando en la unidad. Si hay un problema con el diskette o con la unidad, la luz roja parpadea, incluso aunque el disco deje de girar.

Carga de programas desde diskette

PASO 1 Asegúrese de que su unidad está conectada correctamente. Dele al interruptor de puesta en marcha.

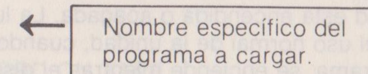
PASO 2 Inserte el diskette en la unidad. La etiqueta del disco debe estar en la parte superior. Inserte el disco por la abertura de modo que la etiqueta sea lo último en introducirse. Vea la pequeña muesca en uno de los lados del diskette (debe estar tapada con un plástico). Esta muesca debe encontrarse a su izquierda mientras Vd. introduce el disco, suponiendo que está Vd. enfrente de la unidad. Asegúrese de que lo hace correctamente.



PASO 3 Cierre la puerta de protección de la unidad tras insertar el disco.

PASO 4 Teclee:

DLOAD" nombre del programa"

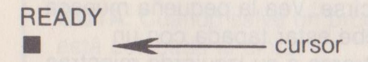


(Para ahorrar tiempo puede Vd. pulsar la tecla de FUNCION 2 y teclear el nombre del programa y las comillas finales).

PASO 5 Pulse la tecla **RETURN.** El disco gira y en la pantalla aparece:

SEARCHING FOR nombre del programa
FOUNDnombre del programa
LOADING

READY
■



PASO 6 Su programa ahora está listo para su uso. Teclee **RUN** y pulse la tecla **RETURN** para ejecutar el programa.

Si la luz roja de la unidad de disco parpadea tras pulsar DLOAD, algo ha ido mal. Escriba:

?DS$ (y pulse **RETURN**)

Para averiguar cual es el problema.

Ejemplos de comandos DLOAD:

DLOAD "*"	Carga el primer programa del disco.
DLOAD "FILES"	Carga un programa llamado FILES.
DLOAD "SOF*"	Carga el primer programa del disco que empiece con las letras SOF.

Formateado del diskette

El formateado prepara para su uso los diskettes en blanco. Cualquier diskette en blanco debe ser formateado, antes de ser usado, utilizando el comando HEADER.

IMPORTANTE: NO FORMATEAR UN DISCO QUE CONTENGA INFORMACION HASTA QUE DESEE BORRARLO TOTALMENTE. EL FORMATEADO BORRA TODO LO QUE HAY EN EL DISCO.

El formato del comando HEADER es:

HEADER "nombre del disco" [,**U**periférico][,**I**i.d.][,**D**drive]

- El nombre es el nombre del disco. Puede darle cualquier nombre hasta un máximo de 16 caracteres.
- El periférico especifica el número de unidad (unidad de disco) y, si se omite, se asume el 8 (unidades de disco Primaria)
- Id. es la letra **I** y cualesquiera otras dos letras o números, como **I**21, **I**R5, etc. Dele al disco cualquier identificación que quiera pero debería darle a cada uno una diferente para evitar confusiones. (Nota: el identificador no puede ser ninguna palabra basic de los caracteres como, **IF, TO, GO,** ETC...)
- Si tiene Vd. una unidad de discos doble (una unidad con dos alojamientos para discos), debería Vd. especificar el número con un 0 o un 1. El número de Drive debe incluirse SIEMPRE en el comando, incluso en el caso de que sea una unidad sencilla (**D**0).

ARE YOU SURE? (¿ESTA SEGURO?)

Tan pronto como pulse **RETURN** tras escribir el comando HEADER, el Commodore 16 le preguntara ARE YOU SURE? Esto es para darle una segunda oportunidad de cambiar de opinión. Si quiere seguir adelante y formatear el disco, teclee **YES** o **Y** (SI) y pulse **RETURN.** Si decide no formatearlo, teclee **NO** o **N** y pulse **RETURN**.

He aquí algunos ejemplos del comando HEADER:

HEADER "CARTAS", 107,D1
HEADER "FINANZAS", U8,IS3,D0

NOTA: Si quiere borrar todos los ficheros de un disco (en vez de formatear un disco en blanco), utilice el comando HEADER sin número de identificación. De este modo borrará todos los ficheros que se encuentren en el disco.

Ahora que ya sabe como formatear un disco, esta preparado para usar discos en los que escribir y salvar programas con su COMMODORE 16 (o cualquier otro ordenador Commodore). La Enciclopedia BASIC al final de este manual incluye más información acerca del comando HEADER.

Salvado de Programas en diskette

Cuando quiera Vd. volver a utilizar un programa hecho por Vd. asegúrese de salvarlo antes de cargar otro programa o de desconectar el COMMODORE 16. Si no lo hace lo perderá.

Cuando cambie un programa ya salvado, deberá igualmente salvarlo de nuevo si quiere conservar la nueva versión.

Cuando vuelve a salvar un programa, esta Vd. reemplazando la versión antigua por la nueva. Si quiere conservar ambas debe darle a la nueva un nombre diferente en el momento de salvarla.

Siga estos pasos para salvar un programa en disco:

PASO 1 Escriba **DSAVE** ''nombre del programa''

PASO 2 Pulse **RETURN.** El ordenador le dará este mensaje una vez salvado:

SAVING 0: ''nombre del programa''
READY
■

Ejemplo:

DSAVE''PROG5'' El nombre del programa puede tener hasta 16 caracteres

Si la luz roja de la unidad parpadea tras la finalización del DSAVE, es que algo ha ido mal. Escriba:

?DS$ (y pulse **RETURN**)

para ver cual ha sido el problema.

Si intenta Vd. salvar el programa en un disco protegido (un disco que no acepta la grabación), debe Vd. apagar la unidad y volverla a conectar de nuevo.

El Comando Directory

Cuando salva Vd. programas en disco, el ordenador actualiza una lista de todos los ficheros grabados en ese disco. Puede Vd. listarla como si fuera un índice para ver lo que hay en un disco utilizando el comando DIRECTORY.

Escriba:

DIRECTORY y, a continuación, pulse **RETURN**
(o pulse la tecla de FUNCION 3)

Tan pronto como pulse **RETURN,** el COMMODORE 16 le listará todo lo que hay en el disco.

También puede listar solo una parte del contenido del directorio:

DIRECTORY''MY*''	Lista todos los ficheros del disco que empiezan con las letras MY.
DIRECTORY''* = P-RG''	Lista todos los ficheros de programas.
DIRECTORY''* = SEQ''	Lista todos los ficheros secuenciales.

CAPITULO 4

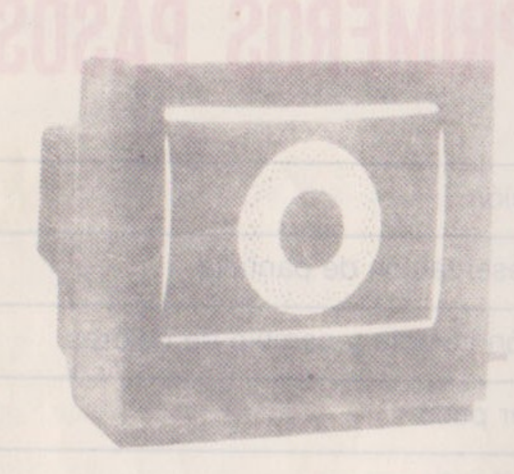
LOS PRIMEROS PASOS

- Introdución
- La representación de pantalla
- Impresión inversa y cambio de colores
- El primer paso
- Corrección de errores
- Ventanas de pantalla

INTRODUCION

El propósito de este capítulo es que empiece a familiarizarse con algunas de las características y capacidades de su COMMODORE 16 y como dar los primeros pasos en la programación de su ordenador.

La Representación en Pantalla



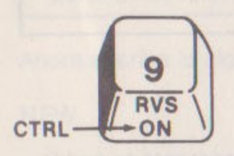
Las entradas y salidas de su COMMODORE 16 son visualizadas a través de la pantalla de su televisor o de un monitor. Cualquier cosa que Vd. escriba, cualquier cosa que su ordenador tenga que decir acerca de lo que Vd. ha escrito, cualquier programa que Vd. cargue... todo aparece en la pantalla de su monitor, sea éste una pantalla de televisión en blanco y negro o color, o sea un monitor en color Commodore. La pantalla de su COMMODORE 16, no importa que tipo de monitor o televisor esté Vd. usando, tiene una anchura de 40 columnas. Esto significa que puede Vd. teclear 40 caracteres desde el extremo izquierdo de la pantalla hasta el estremo derecho. Hay también 25 filas, por lo que una pantalla llena totalmente puede contener 1000 caracteres (25x40). La representación en pantalla del COMMODORE 16 está formada por tres elementos: los caracteres (letras, números y símbolos gráficos), el fondo y el borde. Cuando conecta Vd. su ordenador, los caracteres aparecen en negro, el fondo en blanco y el borde en azul claro. Ya ha visto como cambiar el color de los caracteres utilizando las teclas numéricas conjuntamente con **CTRL** o C=, en el Capítulo 2. Hay algunas otras cosas que debe Vd. saber para hacer que la impresión de caracteres sea más interesante.



Impresión Inversa y Cambio de Colores

No solamente puede Vd. cambiar el color del cursor y de los caracteres que escriba (y su ordenador representa) en la pantalla, sino que puede también invertir los colores de los caracteres y del fondo. Esto significa que, si tiene Vd. caracteres negros sobre fondo blanco y entra Vd. el modo inverso (pulsando **CRTL** y **9**), todo lo que aparezca en la pantalla lo hará en modo inverso. Los caracteres saldrán en blanco, mientras que el fondo debajo de estos caracteres aparecerá en negro. Puede Vd. experimentar cambiando el color de los caracteres y usando la impresión inversa siguiendo estas instrucciones:

PASO 1 Pulse la tecla **CTRL** y **RVS ON.**



PASO 2 Suelte las teclas y pulse la barra espaciadora.

PASO 3 Mantenga pulsada la barra espaciadora todo el rato que quiera. Mientras la mantiene pulsada, aparece una línea del mismo color que las letras a lo largo de la pantalla. Si la línea llega al final de una línea, continúa en la línea siguiente.

PASO 4 Suelte la barra espaciadora (pero no pulse la tecla **RETURN**).

PASO 5 Pulse la tecla **CTRL** y una de las teclas de color (que no esté ya en la pantalla). Tan pronto como lo haga, el cursor cambia al color de la tecla pulsada.

PASO 6 Pulse de nuevo la barra espaciadora. Ahora el ordenador dibuja una línea en un nuevo color. Vaya cambiando los colores con las teclas **CTRL** o C= y las teclas de color. Luego apriete la barra espaciadora para conseguir más líneas de colores diferentes.

PASO 7 Vuelva al modo de impresión normal pulsando **CTRL** y **RVS OFF.** Pulsando la tecla **RETURN** también se puede volver al modo normal.



BREVE INCISO: Puede Vd. teclear letras, números y signos de puntuación en modo inverso. Las letras en modo inverso hacen el papel de títulos de un modo excelente. Puede también utilizarlo para resaltar algunas letras o números de manera especial. Puede escribir también símbolos gráficos en modo inverso por la misma razón. Para poner una raya en su pantalla, seleccione el color que desee y escriba el símbolo — (pulsando **SHIFT** y **E**) a lo largo de toda una línea. Luego pase al modo inverso. Llene la siguiente línea con el mismo signo pero ahora invertido. En la línea siguiente escriba **SHIFT** y **R** (siguiendo en modo inverso). Luego vuelva al modo normal y llene la última línea con **SHIFT** y **R** (en modo normal). Ahora en su pantalla hay una línea en el color que haya Vd. escogido. Quizás sea éste un logro un poco pobre, pero le da a su Pantalla una cierta gracia...

EL PRIMER PASO

Que es un programa, en realidad? Un programador experimentado podría decirle que un programa es una serie ordenada de instrucciones que obligan al ordenador a ejecutar una serie de pasos en vista a conseguir un resultado determinado. De esto, lo que puede Vd. deducir es que no le pregunte a un programador experimentado acercà de que es un programa.

Lo siguiente, aunque solamente tenga una longitud de dos líneas, es un programa. Escríbalo exactamente como aparece aquí. No olvide los números al principio de cada línea ya que son los números de línea. Estos le indican a su ordenador el orden en el que debe leer y ejecutar las líneas del programa. Asegúrese también de pulsar la tecla **RETURN** al final de cada una de las líneas que escriba.

10 PRINT"PRIMER INTENTO"	Esta línea le indica a su ordenador que debe imprimir las palabras PRIMER INTENTO en la pantalla.
20 GOTO 10	Esta línea le indica a su ordenador que debe volver a la línea 10 e imprimir de nuevo PRIMER INTENTO.
RUN	Cuando Vd. escribe esto y pulsa **RETURN**, el ordenador realiza lo indicado en cada una de las líneas.

Pulse la tecla **RUN/STOP** para detener la ejecución del programa.

¿Por que su ordenador ha escrito PRIMER INTENTO tantas veces? PRINT es un comando que le dice a su ordenador que debe imprimir en la pantalla lo que aparece entre comillas. Cuando el ordenador lee esta línea, ejecuta (lleva a cabo) ese mandato. GOTO le dice al ordenador que debe volver a la línea 10 y ejecutar el mandato (**PRINT"PRIMER INTENTO"**). Cada vez que el ordenador llega a la línea 20, ejecuta el mandato GOTO volviendo una y otra vez a la línea 10. Esta repetición se llama bucle. La tecla **RUN/STOP** interrumpe cualquier cosa que el ordenador esté haciendo y devuelve el control al teclado.

AVISO: ¿Quiere ralentizar este programa (o cualquier otro) sin pararlo? Simplemente pulse la tecla C=.

Ahora escriba lo siguiente, pulsando **RETURN** tras cada línea:

NEW ← Esto le indica al ordenador que olvide el último programa y se prepare para otro nuevo.

El ordenador responde con el mensaje:

READY
■ ← No escriba esto; esto se lo dice el ordenador para significar que está preparado para aceptar nuevas órdenes o programas.

10 PRINT"R COMMODORE ▬16" — Pulse **CTRL** y **RVS ON** — Pulse **CTRL** y **RVS OFF**

Observe lo que ocurre cuando pulsa Vd. **RVS ON** y **RVS OFF** entre comillas. Aparecen en la pantalla como caracteres invertidos. Ahora escriba RUN y pulse **RETURN** para ver el programa. Cuando éste se ejecuta, su COMMODORE 16 lee estos caracteres de inversión tal como Vd. los tecleó y activa el modo inverso tal como se le ordenó. El mismo principio se aplica (un carácter invertido en la pantalla como representación de lo que Vd. ha tecleado) para hacer otras cosas dentro de una instrucción PRINT, tales como mover el cursor y cambiar colores, durante la ejecución de un programa.

Intente ahora la misma línea, pero reemplazando **RVS ON** y **RVS OFF** por **FLASH ON** y **FLASH OFF** (las teclas <y>):

10 PRINT"██ COMMODORE ▬16" Pulse **CTRL** y **FLASH ON**
Pulse **CTRL** y **FLASH OFF**

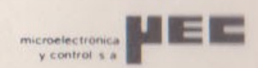
Cuando Vd. ejecute esto, aparecerá en la pantalla COMMODORE 16 pero la palabra COMMODORE parpadeará continuamente, incluso aunque el programa ya haya terminado de ejecutarse. Cuando utiliza Vd. la impresión parpadeante, ésta parpadea continuamente. Estos dos programas de una sola línea le muestran como utilizar cosas tales como la impresión inversa y las letras parpadeantes en las líneas de un programa. Puede Vd. hacer otras cosas (cambiar colores, etc.) de la misma forma.

Comandos de entrada

Se habrá dado cuenta de que teclea algunas cosas como una sola palabra (por ejemplo, NEW), mientras que, en otras líneas, debe Vd. escribir números seguidos por comandos e instrucciones entre comillas. Esto es porque hay dos "modos" que puede Vd. utilizar para comunicarse con su ordenador. Ambos se basan en un lenguaje comprensible para su ordenador. El más manejable (que está incorporado a su COMMODORE 16) se llama BASIC. Su COMMODORE 16 comprende una versión llamada Commodore BASIC 3.5 Los términos (o palabras-clave) del BASIC son el alma de los dos modos. El primero, el MODO INMEDIATO, le dice a su ordenador que ejecute el comando BASIC inmediatamente. Vd. introduce el comando que debe ser ejecutado por el ordenador al pulsar la tecla **RETURN**. Este modo también es conocido como MODO DIRECTO. El otro modo alternativo se conoce como PROGRAMACION o MODO PROGRAMA. El modo de programación presenta números de línea y, cada línea, contiene comandos en BASIC. El programa entero se ejecuta cuando Vd. teclea el comando RUN, ejecutando los mandatos de las líneas de acuerdo con su numeración (la línea más baja primero). Obviamente, el modo de programación es el que se utiliza para escribir programas.

CORRECCIONES DE ERRORES

Los errores están a la orden del día en los ordenadores. Los errores de programación deben ser detectados y corregidos para que un programa se ejecute correctamente. Para eso es para lo que sirve la tecla **HELP**. Un tipo de error más leve, el error de tecleo, puede estropear un



programa, pero el ordenador, no solamente detectará estos errores, sino que le será de gran ayuda a la hora de encontrarlos y corregirlos. Recuerde que para introducir algún cambio (o cualquier otra cosa) ya sea para que la ejecute el ordenador o para que la guarde en la memoria, debe pulsar la tecla **RETURN**.

1. PUEDE VD. CORREGIR UNA LINEA tecleando sobre la misma. Utilice las teclas del cursor para colocarse en el lugar de la línea que desea cambiar. A continuación teclee sobre ella lo que quiere cambiar. Pulse **RETURN** cuando haya terminado.

EJEMPLO:

10 PRINT"SON LAS TRES EN PUNTO"

Si quiere Vd. cambiar la hora a las SEIS, mueva el cursor hasta la T de TRES.

10 PRINT SON LAS ▇RES EN PUNTO

Y, a continuación, escriba SEIS encima de TRES y pulse **RETURN**.

10 PRINT"SON LAS SEIS EN PUNTO"

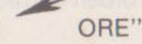
> **NOTA:** Cuando trabaje con líneas de programa numeradas, no necesita estar al final de la línea para pulsar **RETURN.** Su COMMODORE 16 recuerda la línea completa incluso aunque pulse Vd. **RETURN** en medio de ella.

2. PUEDE VD. ABRIR ESPACIOS EN UNA PALABRA O LINEA con la tecla **INST** (Pulse **SHIFT** conjuntamente con **INST/DEL**). Mantenga las dos teclas pulsadas hasta que haya abierto tantos espacios como necesite. (Observe que el cursor permanece en el mismo lugar mientras que los espacios se abren a la derecha). A continuación teclee lo que desea insertar.

10 PRINT"CORE"

Para cambiar esto a COMMODORE, mueva el cursor a la "O" y pulse las teclas **SHIFT** e **INST** hasta que tenga espacio suficiente. No se preocupe de contar los espacios. Puede abrir luego más si ve que no son suficientes.

cursor

10 PRINT"C■  ORE"



Luego escriba las otras letras.

10 PRINT"COMMODORE"

3. **PUEDE VD. BORRAR CARACTERES Y CERRAR EL ESPACIO** con la teclas **DEL** (lo conseguirá pulsando la tecla **INST/DEL** sola). Esta tecla borra los caracteres o los espacios inmediatamente a la IZQUIERDA del cursor.

10 PRINT"PROGRAMA SEMANAL"

Puede Vd. cambiar esto a PROGRAMA DIARIO moviendo el cursor hasta la E de SEMANAL, pulsando una vez la tecla **INST/DEL** y escribiendo DIARIO.

10 PRINT "PROGRAMA S**E**MANAL"

y, pulsando una vez **INST/DEL**:

10 PRINT "PROGRAMA **E**MANAL" — Escriba DIARIO para reemplazar EMANAL y pulse **RETURN.**

4. **PUEDE VD. VOLVER A ESCRIBIR UNA LINEA** en cualquier momento, incluso aunque haya ya ejecutado el programa. Su COMMODORE 16 reemplaza automáticamente la línea anterior por la nueva cuando pulsa Vd. **RETURN** al introducir esta última. La línea anterior sigue apareciendo en la pantalla pero el ordenador la ignora. Cuando tenga Vd. dos instrucciones con el mismo número de línea, su COMMODORE 16 utiliza únicamente la introducida en último lugar. Por ejemplo, en un breve programa utilizando el comando COLOR para cambiar el color del fondo de la pantalla, puede ocurrir un error.

10 COKOR 0,3
20 PRINT"COMMODORE 16"

Pulse la tecla **RETURN** para dejar una línea en blanco y vuelva a escribir la línea 10 correctamente:

10 COLOR 0,3

Ahora la primera línea 10 es reemplazada por la segunda. Puede Vd. comprobarlo escribiendo **LIST,** lo cual sacará un listado línea por línea del programa tal como se encuentra en la memoria del ordenador. Cuando lista Vd. un programa, las líneas aparecen en el orden correcto y las líneas reemplazadas no aparecen:

LIST — **RETURN**
10 COLOR 0,3
20 PRINT"COMMODORE 16" — Esto es lo que aparece en pantalla

A través de este ejemplo puede Vd. ver lo que hace el comando COLOR. Para una explicación más detallada, vea Capítulo 6, o consulte el término COLOR en la Enciclopedia BASIC.

Reemplazar líneas en un programa es también un buen modo de experimentar con su ordenador. Cuando reemplaza Vd. una línea, la nueva línea no tiene por que parecerse en nada a la anterior. Por ejemplo, en vez de corregir el error de COLOR, puede Vd. introducir esta línea:

10 PRINT"COMMODORE 4 X COMMODORE 4 = "

espacio

Ahora ejecute el programa y vea lo que ocurre.

5. **PUEDE VD. BORRAR UNA LINEA NO DESEADA** simplemente tecleando el número de línea y pulsando **RETURN.** El ordenador ignora la línea incluso aunque ésta aparezca en la pantalla. Escriba **LIST** para obtener un listado del programa y asegurarse de que la línea ha desaparecido del programa.

10 PRINT"COMMODORE 4 X COMMODORE 4 = "
20 PRINT"COMMODORE 16"

Ahora escriba:

10 **RETURN**

LIST **RETURN**

El resultado, de acuerdo con su ordenador es:

20 PRINT"COMMODORE 16"

BORRADO DE LA PANTALLA

Hay ocasiones en que la pantalla estará demasiado llena y querrá Vd. reorganizar lo que esta escribiendo o, simplemente, no le gustará lo que hay en la pantalla y querrá que se vaya a donde sea que van las cosas cuando abandonan la pantalla. Hay unos cuantos modos de conseguir esto, dejando intactos los programas en la memoria del ordenador o borrándolos también junto con la pantalla.

Un modo de borrar la pantalla es mantener pulsada la BARRA ESPACIADORA, hasta que haya borrado los 1000 espacios de la pantalla (aproximadamente un par de minutos). Si ha leido esto antes en el Manual del Usuario, es posible que haya tenido la paciencia de borrar la pantalla de este modo, pero también la inteligencia de darse cuenta de que debe haber un modo mejor. He aquí algunos modos mejores:

1. Apriete la tecla de CURSOR ABAJO ↓ hasta que todo el contenido de la pantalla haya desaparecido por la parte superior.

2. Pulse las teclas **SHIFT** y **CLR/HOME** conjuntamente. De este modo borrará la pantalla y vuelve a colocar el cursor en la esquina superior izquierda (posición "home").

3. Escriba:

   SCNCLR y pulse **RETURN**

   éste es un comando del lenguaje BASIC que su ordenador interpreta como una instrucción para limpiar la pantalla.

4. Pulse el BOTON DE RESET. este botón restaura la máquina, vuelve a la pantalla inicial y borra todos los programas de la memoria del ordenador.

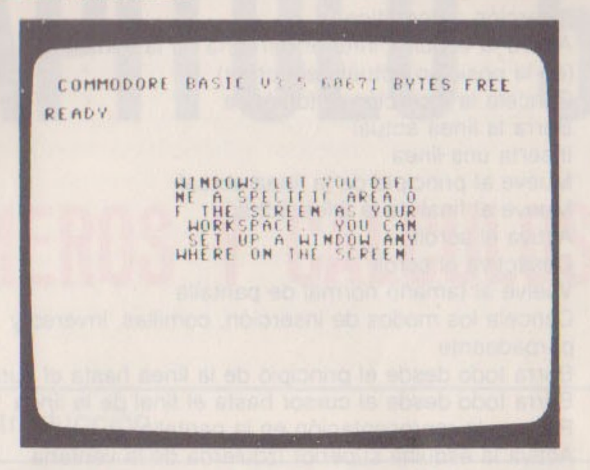
Si quiere Vd. borrar la memoria del ordenador pero no la pantalla, escriba:

NEW y, a continuación, pulse **RETURN.**

Este es un comando del lenguaje BASIC que le dice al ordenador que vacíe la memoria. El 16 en el nombre del COMMODORE 16 se refiere a la cantidad de espacio (16K) de que dispone su ordenador para almacenar programas. Esto es la memoria. Cuando Vd. borra la pantalla con alguno de los tres primeros métodos, borra Vd. los caracteres de la pantalla, pero cualquier cosa que haya en la memoria del ordenador permanece allí. Esto significa que tiene Vd. menos espacio para nuevos programas. Los últimos dos métodos borran lo que hay en la memoria del ordenador, con lo cual tiene Vd. de nuevo la memoria completa para almacenar nuevos programas.



VENTANAS DE PANTALLA



Las ventanas le permiten definir un área específica de la pantalla como espacio de trabajo. Todo lo que Vd. teclee (líneas que escriba, listados de programas, etc.) tras activar una ventana, aparecerá dentro de los límites de la misma, sin afectar a la parte de pantalla exterior a dichos límites. Puede Vd. activar una ventana en cualquier lugar de la pantalla.

Para activar una ventana, siga los siguientes pasos:

PASO 1 Mueva el cursor a la posición de la pantalla que desee Vd. como esquina superior izquierda de la ventana.

PASO 2 Pulse la tecla **ESC** y, tras soltarla, pulse la letra **"T"**.

PASO 3 Mueva el cursor a la posición que desee como esquina inferior derecha de la ventana.

PASO 4 Pulse **ESC** y, tras soltarla, la letra "B". Su ventana está activada.

Todas las salidas de pantalla quedan confinadas a la "caja" que acaba Vd. de definir. Para cancelar la ventana, pulse la tecla **CLEAR/HOME** dos veces. La ventana es entonces borrada y el cursor se coloca en la posición superior izquierda de la pantalla.

Puede manipular la ventana y el texto incluído en ella mediante la tecla **ESC.** Las funciones de edición de pantalla, tales como insertar y borrar textos, scroll y cambiar el tamaño de la ventana, pueden ser efectuadas pulsando la tecla **ESC** seguidas por otra tecla. Para usar una función específica, pulse la tecla apropiada tras pulsar la tecla **ESC.**

TECLA	FUNCION
A	Inserción automática
B	Activa la esquina inferior derecha de la ventana (en la posición actual del cursor)
C	Cancela la inserción automática
D	Borra la línea actual
I	Inserta una línea
J	Mueve al principio de la línea actual
K	Mueve al final de la línea actual
L	Activa el scroll
M	Desactiva el scroll
N	Vuelve al tamaño normal de pantalla
O	Cancela los modos de inserción, comillas, inverso y parpadeante
P	Borra todo desde el principio de la línea hasta el cursor
Q	Borra todo desde el cursor hasta el final de la línea
R	Reduce la representación en la pantalla
T	Activa la esquina superior izquierda de la ventana
V	Scroll hacia arriba de la pantalla
W	Scroll hacia abajo de la pantalla
X	Cancela la función anterior

CAPITULO 5

NUMEROS Y CALCULOS

INTRODUCCION

No tiene Vd. por que ser un genio matemático para comprender y utilizar las capacidades matemáticas de su COMMODORE 16. Además de las operaciones elementales, tales como adición, substracción, multiplicación y división, puede Vd. usar su ordenador para trabajar con funciones avanzadas tales como raíces cuadradas y senos trigonométricos. Aprenderá Vd. los diferentes tipos de variables y el modo de utilizarlas. Su ordenador puede manejar con la misma eficiencia números enteros de una sola cifra, o bien números complejos de hasta 38 cifras expresados en notación exponencial. Puede Vd. efectuar sus cálculos directamente o como parte de un programa. Finalmente, este capítulo le dará una breve explicación de cómo introducir sus propias funciones para que el ordenador las evalúe.

NUMEROS Y OPERACIONES

Puede Vd. utilizar su ordenador como una simple calculadora. Junto a los signos corrientes + y -, su COMMODORE 16 utiliza los signos * para la multiplicación y / para la división y las fracciones.

(Los ordenadores utilizan el signo * para la multiplicación debido a que un ordenador no puede ver la diferencia entre la letra X y el signo matemático x.) Puede Vd. utilizar estos operadores en modo inmediato (sin números de línea) o en líneas de programa. Teclee los números y los operadores en una instrucción PRINT, SIN COMILLAS si quiere que su ordenador efectúe la operación indicada. Si los números y operadores están entre comillas (como, por ejemplo: PRINT "2 + 2"), el ordenador imprimirá exactamente lo que hay escrito en vez de efectuar la operación. Puede Vd. hacer que su ordenador le dé la respuesta de cuantos son dos más dos escribiendo PRINT 2 + 2.

OPERADORES BASICOS MATEMATICOS		OPERADORES BASICOS LOGICOS	
Adición	+	Mayor que	>
Substracción	—	Menor que	<
División y fracciones	/	Igual	=
		Mayor o igual	=> ó >=
Multiplicación	*	Menor o igual	=< ó <=
Exponenciación (Pulse SHIFT y 0)	↑	No igual	<> ó ><

> **NOTA:** Su ordenador no acepta comas formando parte de un número. Por ejemplo, debe Vd. teclear 3.14 en vez de 3,14. Del mismo modo deberá teclear 30359 en vez de 30.359. Si pone Vd. una coma en un número, el ordenador lo entenderá como dos números separados por una coma y leerá 3 y 14 en vez de 3,14.

Fracciones y decimales

Puede Vd. escribir una fracción así: .5
o así: 1/2 Su ordenador efectuará la división.

Si incluye Vd. una fracción en una intrucción PRINT, la respuesta es siempre devuelta en forma decimal o entera. Por ejemplo:

```
PRINT 139/493+5                    RETURN
  5.28194726
```

LA TECLA PI

He aquí un ejemplo que utiliza Pi (3.14159265...), el cual representa la relación entre la longitud de la circunferencia y su diámetro. Utilice este valor pulsando simplemente la tecla Pi [C=] y =):

```
PRINT π/374              RETURN
  8.39998036E-03
```

Notación científica

¿Qué ha querido significar el ordenador con E-03 en la respuesta anterior? Su ordenador representa números decimales en el rango de -999.999.999 hasta 999.999.999. Los números fuera de ese rango (con más de nueve dígitos) son representados automáticamente en notación científica. Puede Vd. introducir números en esta notación y su ordenador los leerá sin ningún problema (ciertamente los convierte con menos problemas que Vd.). La notación científica es utilizada a menudo ya que permite al ordenador representar números largos con menor cantidad de dígitos.

He aquí como aparecería el número 198.505.478 en notación científica:

1.98505478E+8

Sólo aparece un número a la izquierda del punto decimal.

El último número es el número de posiciones que se desplaza el punto decimal.

Para un número inferior a uno, con varios decimales, el signo sería - en lugar de + indicando que el punto decimal se mueve hacia la derecha.

Por ejemplo:

.0003359 = 3.359E-4

Otros ejemplos:

20 = 2E+1	El punto decimal se mueve un dígito a la izquierda.
105000 = 1.05E+5	El punto decimal se mueve cinco dígitos a la izquierda.
.0666 = 6.666E-2	El punto decimal se mueve dos dígitos a la derecha.

CALCULOS

Para efectuar un cálculo, escriba PRINT y, a continuación, el problema matemático sin comillas, del modo siguiente:

10 PRINT 1+2, 2-1
20 PRINT 2*2, 4/2
RUN

```
3    1
4    2
```

PRINT no imprime exactamente lo que Vd. ha escrito en la instrucción. En vez de eso, el ordenador resuelve los cálculos e imprime únicamente las respuestas. Todo lo que debe hacer para utilizar PRINT para hacer cálculos es omitir las comillas. Ahora intente esto:

```
NEW
10 PRINT "2001/2010"
20 PRINT 2001-17
RUN
2001/2010
 1984
```

se deja un espacio a la izquierda de la respuesta para incluir el signo.

Como la línea 10 está entre comillas, el ordenador simplemente imprime el problema como si fuera un texto corriente, exactamente como aparece en las comillas. Observe que no hay espacio para el signo al imprimir la línea 10 mientras que en la respuesta a la línea 20 si lo hay. Ahora mueva el cursor a la línea 10 y cámbiela del modo siguiente:


```
10 PRINT "2001/2010 = ";
  2001/2010
```

no olvide el punto y coma

```
RUN
2001/2010 = .995522388
 1984
```

El espacio antes del punto es el espacio para el signo.

La respuesta de la línea 20 sigue como antes.

Si desea Vd. imprimir el problema y la respuesta deberá escribirlo dos veces, una vez entre comillas y otra sin comillas, tal como muestra la línea 10.

Orden de los cálculos

Puede Vd. efectuar más de un cálculo en una única línea. Intente escribir esto:

PRINT 200*50+5

Es la respuesta que esperaba Vd.? Intente esto:

PRINT 50+5*200

Su COMMODORE 16 siempre efectúa los cálculos en un cierto orden. Los problemas son resueltos de izquierda a derecha y, dentro de esta regla general, algunos tipos de cálculos son resueltos en primer lugar. El orden en el que su ordenador evalúa las expresiones es llamado orden de precedencia.

PRIMERO:	Su ordenador busca los números negativos (no las substracciones, sino sólo los números negativos).
SEGUNDO:	Su ordenador resuelve todos los exponentes.
TERCERO:	Su ordenador resuelve todas las multiplicaciones y divisiones, de izquierda a derecha.
CUARTO:	Su ordenador resuelve las adiciones y substracciones, de izquierda a derecha.

NOTA: Su COMMODORE 16 siempre resuelve primero cualquier parte del problema que se encuentre entre paréntesis. Incluso puede Vd. colocar paréntesis entre paréntesis: 360(12+(A/3)). El contenido del paréntesis más interior se resuelve primero.

A veces puede ser una buena idea poner los números negativos entre paréntesis para una mayor claridad. Por ejemplo, si quiere Vd. multiplicar 45 por -5 escríbalo así: 45*(-5). Su ordenador, de todos modos, puede comprenderlo con paréntesis o sin ellos.

MAS COSAS ACERCA DE LA IMPRESION EN LA PANTALLA

Probablemente habrá notado que en ciertos programas ha escrito Vd. comas en determinados lugares, mientras que en otros ha colocado punto y coma. Su ordenador interpreta las comas y el punto y coma como instrucciones para el espaciado de la salida.

El efecto de la puntuación en el espaciado de una instrucción PRINT trabaja igual tanto si utiliza un texto entre comillas (una cadena), como si lo hace con números que deban ser calculados. Intente escribir este breve programa:

```
NEW
10 PRINT "O", "K"
20 PRINT "O"; "K"
```

Observe que la puntuación aparece FUERA de las comillas.

Cuando ejecute Vd. el programa, la pantalla aparecerá de este modo:

```
O         K        impresión de la línea 10
OK                 impresión de la línea 20
```

Si la línea 10 y la 20 son casi idénticas, ¿por qué hay esa diferencia en lo que imprimen en la pantalla? La única diferencia es debida a la puntuación entre los elementos que imprime el programa.

Cuando utiliza Vd. una coma para separar elementos en una instrucción PRINT, los elementos se imprimen con una separación de algunos espacios. Cuando utiliza punto y coma, los elementos se imprimen uno junto a otro.

Como recordará, su ordenador tiene una pantalla de 40 columnas de longitud. Estas columnas se dividen en cuatro áreas de 10 espacios, llamadas ZONAS DE IMPRESION. Cuando utiliza Vd. una coma para separar los elementos en la instrucción, su COMMODORE 16 imprime el primer elemento en la primera zona de impresión, el segundo elemento en la segunda zona, etc. Las comas trabajan como el tabulador en una máquina de escribir.

Si intenta Vd. imprimir más de cuatro elementos separados por comas, su ordenador automáticamente pasa a imprimir en la siguiente línea.

Por ejemplo:

```
PRINT "A", "B", "C", "D", "E", "F"
```


espacia las letras en la pantalla de este modo:

```
           1     11     21     31     40 COLUMNA
LINEA 1    A     B      C      D
      2    E     F
```

Cuando utiliza Vd. punto y coma para separar elementos en una instrucción PRINT, su COMMODORE 16 ignora las zonas de impresión y coloca todos los elementos uno tras otro:

```
PRINT "A"; "B"; "C"; "D"; "E"; "F"
```

imprime esto:

```
ABCDEF
```

Si el primer elemento del PRINT es de 12 letras de largo y el segundo elemento está separado por una coma, he aquí lo que ocurre:

```
PRINT "ABCDEFGHIJKL", "M"
```

imprime esto:

```
ZONA 1          ZONA 2          ZONA 3
ABCDEFGHIJKL                    M
```

NOTA: Algunas veces, escribirá Vd. alguna línea especialmente larga, como por ejemplo:

```
10 PRINT "ME GUSTA TU MODO DE TOCAR MI TECLADO. VENDRAS
POR AQUI A MENUDO?"
```

Observará que mientras escribe Vd. esto, se sale de la línea. Pero continue escribiéndolo. El COMMODORE 16 automáticamente se mueve a la línea siguiente y continúa imprimiendo hasta que la línea esté terminada. Puede Vd. teclear hasta 80 caracteres en una línea de programa (dos líneas completas de 40 columnas).

Ejecute ahora este programa de una única línea. El mensaje se imprimirá en dos líneas. Si el mensaje es mayor de una línea, su COMMODORE 16 pasa a la línea siguiente. El ordenador considera que la instrucción está terminada cuando Vd. pulsa la tecla **RETURN,** no cuando llega Vd. al final de la línea. Utilizará esto cuando empiece a trabajar con su COMMODORE 16.

VARIABLES

El ejemplo 36*(12+(A/3)) muestra una de las más poderosas capacidades de un ordenador. Cuando, en un problema matemático, tiene Vd. una letra en vez de un número, está utilizando una VARIABLE. Una variable representa un valor:

```
10 A = 3
20 PRINT "TOTAL:"; A*4
```

Si ejecuta Vd. este programa, en la pantalla aparecerá:

TOTAL: 12

Hay tres tipos de variables que puede Vd. utilizar:

TIPO	SIMBOLO	DESCRIPCION	EJEMPLO	VALORES DE EJEMPLO
Coma flotante		números reales (decimales) o enteros.	X, AB, T4	23.5, 12 1.3E+2
Entera	%	números enteros	X%, A1%	15, 102, 3
Cadena	$	letras, números y todo tipo de caracteres entre comillas	X$, MS$	"TOTAL:" "DIA 1" "CBM"

Cada vez que desee que una variable sea leída como un número entero, el nombre de esa variable debe terminar con el signo %. Una variable que contenga un texto DEBE terminar con el signo $ como parte del nombre de la variable. Si no se encuentra este símbolo, el ordenador esperará encontrar un número en coma flotante. Una variable sin ninguno de los dos símbolos (% ó $) se lee como un número en coma flotante (un número "real"). Las variables enteras (números enteros) son un subgrupo de las variables de coma flotante. Son números sin cifras decimales. Asegúrese de utilizar siempre el tipo correcto de variable. Si intenta hacer algo como asignar una palabra a una variable entera, su programa no funcionará. Este programa le muestra que variable puede y no puede ser utilizada en una situación dada y puede ver lo que ocurre cuando utiliza diferentes tipos de datos:


```
10 PRINT"INTRODUZCA UN NUMERO"
20 INPUT X%
30 PRINT"LEO SU NUMERO COMO";X%
40 PRINT"ES BONITO, VERDAD"
50 END
```

Cuando ejecute este programa, intente introducir estos valores (uno cada vez que lo ejecute) cuando el ordenador se lo pida y vea lo que ocurre:

UN QUINTO
.043
10

FUNCIONES NUMERICAS

Incluidas en el lenguaje BASIC de su ordenador hay funciones numéricas tales como cálculos avanzados que podría Vd. encontrar en una calculadora científica (seno, coseno, tangente, etc.).

La mayor parte de funciones puede utilizarse escribiendo el nombre de la función y, entre paréntesis, el número que debe ser operado por la fórmula:

FUNCION (X)

Por ejemplo, para hallar el seno de una variable, deberá Vd. escribir:

PRINT SIN(X)

siendo X el número que Vd. desee.

Puede también incluir una de las funciones en una línea de programa como muestra el siguiente ejemplo:

```
10 FOR X=1 TO 5
20 PRINT "LA RAIZ CUADRADA DE"; X; "ES"; SQR(X)
30 NEXT X
```

Encontrará una lista completa de las funciones numéricas en la Enciclopedia BASIC, al final de este manual.

> **NOTA:** La mayor parte de comandos tienen una abreviatura que puede Vd. escribir en vez de teclear el nombre BASIC completo. El COMMODORE 16 interpreta la abreviatura exactamente como si hubiera leído el nombre completo. Un ejemplo de abreviatura puede ser el siguiente:
>
> L y, a continuación, **SHIFT** O
>
> El ordenador interpreta esto como si hubiera Vd. escrito LOAD. Las abreviaturas son muy útiles para ahorrar tiempo. En la Enciclopedia BASIC hay una lista completa de las abreviaturas aceptadas.

Funciones definidas por el usuario

Un modo efectivo de utilizar la capacidad matemática de su ordenador consiste en crear funciones definidas por el usuario. Estas funciones son extremadamente útiles en los cálculos y son fáciles de incorporar a su COMMODORE 16. Las funciones definidas por el usuario le permiten programar una fórmula y, a continuación, hacer que su ordenador sustituya en ella los valores que desea Vd. calcular. Esto puede ser utilizado para diferentes propósitos.

He aquí una instrucción que utiliza una función definida por el usuario para calcular el valor de una secante:

```
10 DEF FNS(X) = 1/COS(X)
```

Esto nos da el valor de la secante de cualquier número introducido en lugar de X. FNS es el nombre de la función definida en esta instrucción. El apéndice C contiene una tabla de funciones matemáticas no incluidas en el lenguaje BASIC de su ordenador.

La utilización de este tipo de funciones ahorra espacio en la memoria cuando debe Vd. utilizar una función más de una vez y hace que sus programas sean más fáciles de leer y comprender.

CAPITULO 6

GRAFICOS Y COLOR

- Caracteres gráficos
- Animación con caracteres
- Control del color
- Gráficos de alta resolución
- Puntos, líneas y rótulos
- Cuadrados, círculos, polígonos y dibujos
- Gráficos multicolores

CARACTERES GRÁFICOS

Recordará el Capítulo 2 que cada tecla de letra contiene dos caracteres gráficos diferentes que, con los que contienen las teclas @, −, *, y £, hacen un total de 62 caracteres gráficos. Para imprimirlos, debe pulsar las teclas **SHIFT** ó C= mientras pulsa la tecla correspondiente al signo que desee.

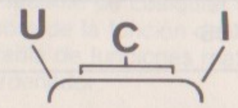
Cuando su COMMODORE 16 no está en modo mecanográfico, la pulsación de la tecla **SHIFT** conjuntamente con otra tecla dará como resultado el carácter situado a la derecha de la tecla. Estos caracteres incluyen las figuras de la baraja francesa, una bola sólida y otra hueca y una serie de líneas y caracteres de conexión que puede Vd. utilizar para hacer diferentes dibujos en la pantalla.

La tecla C= pulsada con una letra produce siempre el carácter dibujado en la parte izquierda de la tecla. Los gráficos del lado izquierdo consisten en un surtido de barras, cuadrados, líneas y bloques muy útiles para dibujar tablas y diagramas.

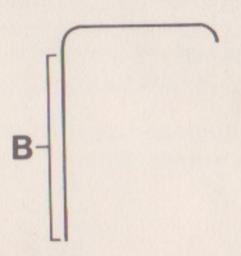
Elija una carta, cualquier carta

He aquí un ejemplo que le ayudará a hacerse una idea más clara de como usar los caracteres gráficos para hacer una representación en la pantalla. Siga estas instrucciones para crear un naipe de la baraja, en este caso el seis de corazones.

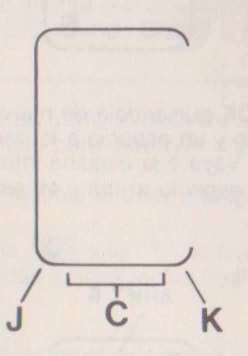
Primero, cambie el color del cursor al rojo. Pulse conjuntamente las teclas **CTRL** y "3" para cambiar el color del cursor.



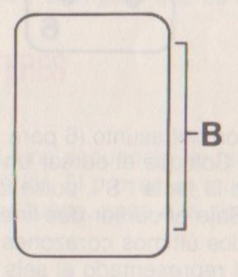
Ahora es el momento de dibujar la parte superior del naipe. Pulse la tecla **SHIFT LOCK** y, de ese modo, quedará fijada. Pulse "U" y, a continuación, "C" cinco veces. Por último, pulse la letra "I".



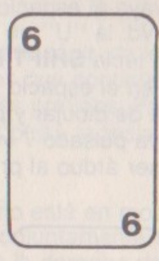
Para dibujar el lado izquierdo, vaya al espacio directamente inferior al que contiene la curva (donde pulsó Vd. la "U" en el paso anterior). Manteniendo todavía pulsada la tecla **SHIFT LOOCK** pulse la letra "B", a continuación coloque el cursor en el espacio inmediatamente debajo del segmento de la línea que acaba de dibujar y pulse de nuevo la "B". "Repita lo mismo hasta que haya pulsado 7 veces la letra "B". (El manejo de las teclas del cursor puede ser árduo al principio pero le será más fácil con la práctica).



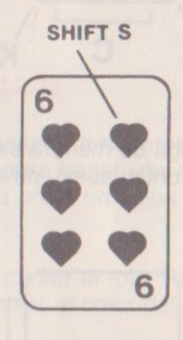
Para dibujar la línea inferior haga de manera similar a como lo hizo con la línea superior. Pulse "J" y, a continuación, pulse la "C" cinco veces terminado con la tecla "K".



Para dibujar la línea derecha hágalo del mismo modo que con la izquierda, pulsando la letra "B" un total de 7 veces (colocando el cursor en el lugar apropiado).



Libere la tecla **SHIFT LOCK** pulsándola de nuevo. En la esquina superior izquierda (un espacio abajo y un espacio a la derecha de la línea redondeada) pulse el "6". Vaya a la esquina inferior derecha, posicionando el cursor un espacio arriba y un espacio a la derecha y teclee otro "6".



Ahora vayamos al corazón del asunto (6 para ser exactos). Pulse la tecla **SHIFT LOCK** de nuevo. Coloque el cursor un espacio abajo y un espacio a la derecha del 6. Pulse la tecla "S", pulse la BARRA ESPACIADORA y de nuevo pulse la "S". Baje el cursor dos líneas y repita la misma operación. Coloque los dos últimos corazones dos líneas más abajo. Ahora tiene Vd. un naipe oficial representado el seis de corazones. No le servirá de gran cosa jugando al black jack, desde luego. Si quiere Vd. mejorar su mano, cree un par de cartas Vd. mismo.

> **NOTA:** Cuando la tecla **SHIFT LOCK** está pulsada, puede Vd. Pulsar **RETURN** sin obtener un mensaje SYNTAX ERROR como respuesta. Incluso aunque tenga Vd. en la linea caracteres que no sean comandos que el ordenador pueda comprender, el ordenador interpreta la tecla **RETURN** simplemente como una orden para saltar al principio de la línea siguiente cuando la tecla **SHIFT** está pulsada. El ordenador no intenta leer o interpretar lo que se haya tecleado en lenguaje BASIC cuando **SHIFT** o **SHIFT LOCK** están pulsadas.

Puede Vd. utilizar las teclas de gráficos para realzar las salidas impresas. Por ejemplo, he aquí el método para subrayar una palabra o una línea:

Primero, mueva el cursor a la línea inferior a la que desea subrayar.

A continuación pulse la tecla C= y la "T", la cual imprime una línea de subrayado. Mantenga pulsadas ambas teclas hasta que esté subrayada la porción del texto deseada.

El propósito de esta sección es mostrarle como pueden ser manipulados los símbolos gráficos de su COMMODORE 16 para crear diferentes formatos y figuras utilizándolos en un sentido más práctico. Además de los 62 caracteres gráficos que tiene Vd. disponibles, también puede utilizar los inversos de esos caracteres. Ahora que tiene Vd. ya una idea general de como utilizar los símbolos gráficos para conseguir diferentes formatos, debería experimentar con ellos y averiguar que es lo que puede conseguir.

ANIMACION CON CARACTERES

Las películas son, en realidad, una secuencia de fotos fijas. Cada foto es ligeramente diferente a la anterior. El proyector muestra cada figura durante un tiempo muy corto y, luego, pasa a la siguiente. La escena se convierte en animada.

La animación en el ordenador trabaja del mismo modo. Primero el ordenador dibuja una figura y, luego, cambia esta figura ligeramente. Su COMMODORE 16 es bastante rápido moviendo objetos suavemente por la pantalla en sus juegos y en sus programas de prácticas. Una película funciona a razón de 30 imágenes por segundo. Los cambios son lo bastante rápidos como para provocar una ilusión óptica y crear sensación de movimiento. El único modo en el que puede alcanzar la velocidad para alcanzar esa ilusión en su ordenador es utilizando un programa para dibujar una imágen, esperar una fracción de segundo y, enseguida, cambiar a otra imágen.

Para conseguir que el programa cree dibujos puede Vd. usar la instrucción PRINT con los caracteres gráficos. El tipo más simple de animación requiere la alternancia de dos caracteres para dar el efecto de movimiento.

Este programa simula animación alternando el círculo (**SHIFT** y Q) y el corazón (**SHIFT** y S). Si utiliza Vd. su imaginación puede considerar esto como un latido algo mediocre.

> **NOTA IMPORTANTE:** Cada vez que el texto se refiera a **SHIFT** o **C=**, debe ser tecleado AL MISMO TIEMPO que la tecla siguiente al introducir el programa, ya que no ocurre nada cuando las dos teclas son pulsadas por separado.

Recuerde escribir NEW y pulsar **RETURN** antes de introducir cada nuevo programa. También debe pulsar **RETURN** para introducir cada línea en todos los programas.

```
10 PRINT"[HOME][SHIFT]Q"
20 FOR L=1 TO 100
30 NEXT L
40 PRINT"[HOME][SHIFT]S"
50 FOR M=1 TO 200
60 NEXT M
70 GOTO 10
```

> **NOTA IMPORTANTE:** En el listado anterior se han colocado entre corchetes ([ ]) las teclas **SHIFT**, **C=** (representada como una C seguida de un =), y ESPACIO para facilitar el trabajo de teclear el programa. De hecho al listar el programa éste aparecerá en la pantalla de la forma siguiente:

```
10 PRINT"■■"
20 FOR L=1 TO 100
30 NEXT L
40 PRINT"■■"
50 FOR M=1 TO 200
60 NEXT M
70 GOTO 10
```

En los listados sucesivos que aparezcan en este manual se seguirá este procedimiento para evitar problemas al usuario a la hora de entrar programas. Cuando aparece la indicación **SHIFT** o **C=** estas teclas



deben pulsarse juntamente con la tecla que sigue a la indicación, así en el programa anterior en la línea 10 donde se indica **(SHIFT Q)** debe pulsarse la tecla **SHIFT** junto con la Q. En el caso de las indicaciones [**HOME**] o [**CLR**] debe pulsarse la tecla CLR/HOME sin la tecla **SHIFT** o con ella, respectivamente (esto se hace extensivo a otras teclas como por ejemplo: INST/DEL, F1/F2, etc...)

Verá Vd. la obvia limitación de la animación alternando dos caracteres, cuando ejecute este programa. Para pararlo, pulse la tecla **RUN/STOP.**

Para conseguir un efecto más interesante puede Vd. construir una figura con varios caracteres gráficos y, luego, cambiar algunos de éstos mientras los otros permanecen en el mismo sitio. Esto hace el efecto de una parte de una figura mayor moviéndose, como en el siguiente programa:

```
10 PRINT"[HOME][SHIFT]M[SHIFT]W[SHIFT]N"
20 PRINT"[ESPACIO][C=]+[ESPACIO]"
30 PRINT"[SHIFT]N[ESPACIO][SHIFT]M"
40 FOR L=1 TO 300:NEXT L
50 PRINT"[HOME][ESPACIO][SHIFT]W[ESPACIO]"
60 PRINT"[C=]T[C=]+[C=]G"
70 PRINT"[ESPACIO][C=]G[C=]G"
80 FOR L=1 TO 300:NEXT L
90 GOTO 10
```

Escriba **RUN** y pulse **RETURN.**

En los dos anteriores ejemplos de animación la figura ha permanecido estacionaria en un área de la pantalla. El siguiente paso consiste en mover la figura animada. La función TAB nos ayudará cuando deseemos mover objetos desde el borde izquierdo. (La función TAB es explicada en detalle en la Enciclopedia BASIC). El siguiente programa representa una serpiente reptando por la pantalla.

Recuerde que **SHIFT** y la tecla siguiente deben ser pulsadas juntas.

```
5 FOR A=0 TO 30
10 PRINT"[SHIFT][CLR]"
20 PRINT TAB(A)"[SHIFT]U[SHIFT]I[SHIFT]U[SHIFT]I"
30 PRINT TAB(A)"[SHIFT]K[SHIFT]J[SHIFT]K[SHIFT]J
  [SHIFT]Q"
40 FOR L=1 TO 100:NEXT L
50 PRINT"[SHIFT][CLR]"
60 PRINT TAB(A+1)"[SHIFT]I[SHIFT]U[SHIFT]I[SHIFT]U
  [SHIFT]W"
70 PRINT TAB(A+1)"[SHIFT]J[SHIFT]K[SHIFT]J[SHIFT]K"
80 FOR L=1 TO 100:NEXT L
90 NEXT A
```




Utilizando caracteres como la pelota (**SHIFT Q**), Puede Vd. disfrutar de videojuegos en la pantalla. Para mover una pelota, simplemente bórrela y colóquela en una nueva posición tal como hace este programa:

```
10 PRINT"[SHIFT][CLR]"
20 PRINT"[ESPACIO][SHIFT]Q[CRSR IZQUIERDA]";
30 FOR L=1 TO 50:NEXT L
40 GOTO 20
```

Cuando ejecute el programa, recuerde pulsar la tecla **RUN/STOP** cuando quiera detener el movimiento de la pelota.

CONTROL DEL COLOR

Pueden colocarse colores distintos en cada parte de la pantalla. El borde puede tener un color, el fondo otro diferente y cada carácter puede tener el suyo propio. Ya sabe como activar el color de los caracteres utilizando el teclado. Puede cambiar el color de las otras áreas de la pantalla utilizando el comando del lenguaje BASIC, COLOR.

Por ejemplo, puede poner en rojo el borde de la pantalla tecleando el comando **COLOR 4, 3** y pulsando la tecla **RETURN.** El número 4, en el comando, significa el área del borde y el número 3 es el número correspondiente al color rojo (el mismo que hay en la tecla marcada RED-ROJO).

Pruebe a escribir **COLOR 0, 7** y pulse **RETURN.** El fondo de la pantalla pasa al color azul. El número 0 significa el fondo de la pantalla, mientras que el 7 es el color azul (también el mismo del teclado).

El primer número tras la pantalla **COLOR** especifica el área de la pantalla que desea Vd. cambiar. Esta tabla contiene los números de control de cada área. Aprenderá a utilizar las áreas 2 y 3 cuando llegue a los gráficos multicolores más adelante, en este mismo capítulo.

Números de las áreas de pantalla

AREA	NOMBRE DEL AREA
0	Fondo
1	Caracteres
2	Multicolor-1
3	Multicolor-2
4	Borde

El segundo número tras **COLOR** selecciona el color que desea Vd. poner en el área de pantalla especificada. Los números de color corresponden a los números de las teclas en el teclado.

Números de color

NUMERO	COLOR	NUMERO	COLOR
1	NEGRO	9	NARANJA
2	BLANCO	10	MARRON
3	ROJO	11	VERDE AMARILLO
4	CIAN	12	ROSA
5	PURPURA	13	VERDE AZUL
6	VERDE	14	AZUL OSCURO
7	AZUL	15	AZUL OSCURO
8	AMARILLO	16	VERDE CLARO

Cada color tiene también un nivel ajustable de brillo, llamado la luminosidad. Puede Vd. ponerle un número desde 0 (más oscuro) hasta 7 (más brillante) tras el número de color. Escriba **COLOR 4, 3, 0** y pulse **RETURN.** El borde cambia a rojo oscuro. Escriba **COLOR 4, 3, 7** y el borde cambia a rojo brillante.

En resumen, el comando COLOR queda del modo siguiente:

COLOR, área, color, luminosidad

He aquí un rápido programa que le muestra todos los colores del COMMODORE 16.

Primero escriba NEW y pulse **RETURN.**

```
10 COLOR0,7,7
20 FORM=0TO7
30 FORN=1TO2
40 FORL=1TO16
50 PRINT"[CTRL][RVS ON]";
60 READA
70 COLOR1,A,M
80 PRINT"[ESPACIO][ESPACIO]";
90 NEXTL
100 PRINT
110 RESTORE
120 NEXTN,M
130 COLOR1,2,4
200 DATA7,14,4,13,6,16,11,8,10,9,3,12,5,15,2,1
```

Cuando ejecute este programa, el fondo de la pantalla cambia a azul claro y el espectro de colores del COMMODORE 16 se muestra en cada nivel de luminosidad. Observará que el negro es igual en todos los niveles de luminosidad.

NOTA: Al igual que en la mayor parte de términos gráficos del BASIC vistos en este capítulo, COLOR puede referirse indistintamente a una intrucción o a un comando.

EL COMANDO GRAPHIC

Los gráficos que haya visto ahora utilizan el teclado sin hacer uso real de las capacidades del ordenador. El lenguaje BASIC de su COMMODORE 16 contiene comandos para dibujar figuras y formatos por medio de programas. Para usar los comandos relativos a gráficos, debe introducir un nuevo modo, el modo **GRAFICO.** El modo gráfico puede ser considerado el modo de dibujo, ya que todos los comandos de dibujo están "activados". Vd. no puede utilizar estos comandos hasta que no especifique exactamente el modo gráfico que desea utilizar. Vd. especifica ese modo mediante el comando **GRAPHIC.** Hay tres modos diferentes: texto normal, gráfico de alta resolución y modo gráfico multicolor. Con el comando **GRAPHIC,** puede incluso tener en la pantalla una parte en modo texto y otra en modo gráfico y puede escribir en una parte y dibujar en el resto. El comando para entrar este nuevo modo es **GRAPHIC.**

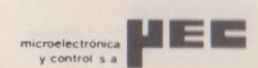
En general, el comando **GRAPHIC** tiene este formato:

GRAPHIC modo, borrado (opcional)

Número de modo	Efecto
0	Texto
1	Gráficos de alta resolución
2	Gráficos de alta resolución + texto
3	Gráficos multicolores
4	Gráficos multicolores + texto

Número de borrado	Efecto
0	No borra la pantalla
1	Borra la pantalla

Para pasar a los gráficos normales (o modo texto) a alta resolución, simplemente escriba el comando **GRAPHIC 2, 1** y pulse **RETURN.** La pantalla queda en blanco y el cursor aparece cerca del final de la



pantalla. La pantalla de su COMMODORE 16 se divide en dos secciones separadas: la superior para gráficos y las cinco líneas inferiores para texto. Si no desea estas últimas, puede usar el comando **GRAPHIC 1, 1** pero entonces no podrá ver ninguno de los comandos que escriba. Puede Vd. regresar del modo gráfico al modo texto usando el comando **GRAPHIC. GRAPHIC 0** retorna la pantalla al modo texto, mientras que **GRAPHIC 2** regresa a la alta resolución sin borrar la pantalla.

Hay otro modo para borrar la pantalla de alta resolución. El comando **SNCLR** limpia la pantalla sin salir del modo gráfico. Cada vez que utilice Vd. gráficos de alta resolución, el ordenador reserva 10K de memoria para la pantalla. Esta memoria se coge del área para programas BASIC. Cuando esté Vd. utilizando gráficos, puede Vd. solicitar esta memoria utilizando el comando **GRAPHIC CLR.**

GRAFICOS DE ALTA RESOLUCION

La pantalla de su COMMODORE 16 contiene 25 líneas de 40 caracteres cada una, en total, 1000 posiciones en la pantalla. Cada carácter está formado por puntos, con 8 filas de 8 puntos cada una formando cada carácter. La pantalla tiene un total de 320 puntos en cada línea y 200 líneas de puntos, en total, 64.000 puntos. Los gráficos de alta resolución de su COMMODORE 16 le permiten controlar cada uno de esos puntos.

Utilizando gráficos en modo normal, tiene Vd. limitado el control sobre los puntos individuales. Dibujar un naipe es un bonito y simple ejercicio, pero su creación está limitada a la utilización de los caracteres (letras, símbolos gráficos, etc.) de su teclado. Puede Vd. hacer una gran cantidad de figuras, pero solamente una fracción de las que podría hacer si pudiera controlar cada punto aisladamente. La capacidad de los gráficos de alta resolución de su COMMODORE 16 le permite hacer eso. La resolución se refiere a la precisión y al control que Vd. posee en el dibujo. Con los gráficos de alta resolución puede Vd. utilizar comandos que le permiten dibujar y borrar puntos, líneas, círculos y otras figuras.

Hay un límite a los gráficos de alta resolución (abreviadamente hi-res). Su ordenador sólo puede utilizar dos colores en cada posición de carácter. Es decir, cada grupo de 8 x 8 puntos (en donde iría un carácter) está limitado a dos colores (el del carácter y el del fondo en esa posición). Puede utilizar Vd. diferentes colores para cada posición de caracteres, pero sólo dos colores en cada posición. Otro modo gráfico que cubriremos más adelante, le permite utilizar diferentes colores por cada posición de carácter a costa de perder algo la resolución que posee el modo gráfico de alta resolución.

He aquí un programa que utiliza algunas de las capacidades del modo gráfico de alta resolución de su ordenador y, en particular, el comando GRAPHIC. Limpie la memoria del ordenador con el comando **NEW,** así se asegurará de que no haya líneas sobrepuestas de programas anteriores y, luego, escriba:

```
10 COLOR 0,1
20 GRAPHIC1,1
30 FORL=2TO16
40 COLOR1,L,2
50 DRAW1,0,L*12TO319,L*12
60 DRAW1,L*18,0TOL*18,199
70 NEXTL
80 FORL=1TO5000:NEXT
90 COLOR1,2,7
100 GRAPHIC0
```

Observe que el color cambia cerca de las intersecciones. Esto es debido a las limitaciones de los gráficos hi-res, con demasiados colores, demasiado juntos.

Puntos, líneas y rótulos

Escriba los comandos **GRAPHIC 2, 1: DRAW 1,0,0** y pulse **RETURN.** Mire de cerca la esquina superior izquierda de la pantalla. El COMMODORE 16 dibujará en ella un punto blanco. El comando DRAW puede ser usado para dibujar, en cualquier lugar de la pantalla, un punto, una línea o una figura.

Algunas formas del comando DRAW son:

COMANDO	RESULTADO
DRAW color, columna, fila	PUNTO
DRAW color, columna, fila TO columna, fila	LINEA
DRAW color TO columna, fila	LINEA DIBUJADA DESDE EL ULTIMO PUNTO

El color es 0 para utilizar el color del fondo, 1 para el de los caracteres. Cualquier cosa dibujada en el color del fondo (0) es lo mismo que borrar el color de los caracteres en ese punto.

En el comando DRAW el primer número es 1 (dibujar un punto) o 0 (borrar

un punto). Los dos números siguientes son las coordenadas de la posición del punto. Así, si quiere Vd. dibujar un punto en la columna 17, fila 20, deberá escribir **DRAW 1, 17, 20.** Para borrar el mismo punto, deberá poner **DRAW 0, 17, 20.**

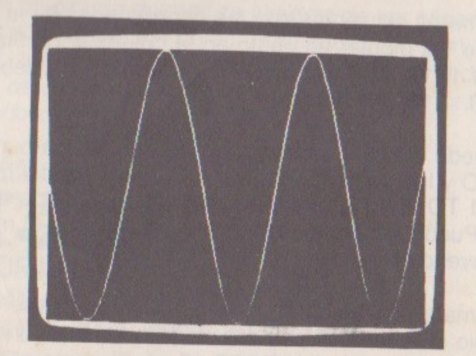
El comando DRAW puede también dibujar una línea entre dos puntos. Simplemente añada TO y las coordenadas del otro extremo, del modo siguiente: **DRAW 1,1,1 TO 100,100.** Esto dibuja una línea desde el punto 1,1 hasta el 100,100. Puede Vd. borrar esta línea escribiendo el mismo comando, pero sustituyendo por 0 el 1 inmediato a DRAW.

Si ha utilizado Vd. en matemáticas los dibujos de gráficos puede encontrarse al principio un poco confuso utilizando el ordenador, el sistema de coordenadas de su COMMODORE 16 es diferente al que ha utilizado Vd. antes. En matemáticas, el punto 0,0 debería estar en el centro o en la esquina inferior izquierda de la pantalla pero, en su ordenador, este punto se encuentra en la esquina superior izquierda. Se acostumbrará a este sistema a medida que lo vaya practicando.

Una vez que haya dibujado un punto o una línea en la pantalla, puede Vd. dibujar una línea desde ese punto hasta cualquier otro, de este modo: **DRAW 1 TO 150,50.** Esto dibuja una línea desde el último punto dibujado hasta la columna 50, fila 150. Si su programa utiliza una serie de comandos **DRAW,** puede colocar el primer punto en una posición de la pantalla utilizando el comando **LOCATE,** por ejemplo **LOCATE 100,100,** para volver inmediatamente a la primera posición.

Este programa dibuja una curva basada en la función seno.

```
10 COLOR 0,1
20 COLOR 1,2
30 GRAPHIC 1,1
40 LOCATE 0,100
50 FOR X=1 TO 319
60 Y=INT(100+99*SIN(X/20))
70 DRAW 1 TO X,Y
80 NEXT X
90 FOR L=1 TO 5000
100 NEXT L
110 GRAPHIC 0
```



No escriba NEW tras ejecutar el último programa. Para trazar el dibujo de un modo diferente, cambie la línea 70 a:

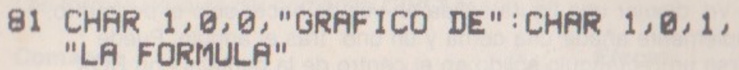
70 DRAW 1,X,Y

Este programa dibuja la misma curva utilizando puntos en vez de líneas.

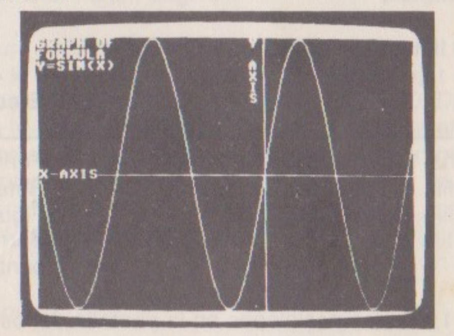
El comando CHAR

Los gráficos más fáciles de comprender y de utilizar si los rotula. Puede Vd. utilizar el comando CHAR para mezclar textos en medio de un dibujo en alta resolución. Por ejemplo, la instrucción **CHAR 1,0,5, "HOLA"** pone la palabra HOLA en la sexta línea a partir de la parte superior izquierda de la pantalla (la primera línea es "0"). El primer número tras la palabra CHAR es 1 (para dibujar) o 0 (para borrar). Los dos números siguientes son la columna y la fila donde aparecerá el texto.

Deje los dos últimos programas en el ordenador, sin escribir NEW. Añada estas líneas:



```
81 CHAR 1,0,0,"GRAFICO DE":CHAR 1,0,1,
   "LA FORMULA"
82 CHAR 1,0,2,"Y=SIN(X)"
83 DRAW 1,0,100 TO 319,100,189,0 TO 189,199
84 CHAR 1,0,12,"EJE-X":CHAR 1,22,0,"E"
85 CHAR 1,22,1,"J":CHAR 1,22,2,"E"
86 CHAR 1,22,4,"Y"
```



CUADRADOS, CIRCULOS, POLIGONOS Y DIBUJOS

Utilizando el comando DRAW puede dibujar figuras trazando algunos puntos o líneas. Para dibujar un cuadrado, puede utilizar el comando **DRAW 1,0,0 TO 100,0 TO 100,100 TO 0,100 TO 0,0** para conectar cuatro puntos con cuatro líneas (uniendo cada vértice del cuadrado) o puede simplemente usar el comando **BOX**.

Dibujo de rectángulos

Uno de los comandos gráficos de su COMMODORE 16 hace más fácil el dibujar cuadrados y otras figuras rectangulares. El comando **BOX** le permite marcar los puntos de dos esquinas opuestas del cuadrado. Para duplicar la misma figura del ejemplo anterior, use simplemente **BOX 1,0,0,100,100.** El número 1, nuevamente significa que desea Vd. dibujar y no borrar. Los cuatros números siguientes son las coordenadas de los vértices opuestos del cuadrado, (0,0) en la esquina superior izquierda y (100,100) cerca del centro de la pantalla.

El comando **BOX** puede formar cualquier rectángulo simplemente cambiando las esquinas. Incluso puede Vd. rotar el dibujo especificando un ángulo (en grados) tras la última coordenada, del modo siguiente: **BOX 1,50,50,100,100,45.** La figura ha sido rotada 45 grados en el sentido de las agujas del reloj formando algo parecido al palo de diamantes de la baraja francesa.

Si quiere Vd. dibujar una figura sólida en vez de hacer solo el perímetro, debe simplemente añadir una coma y un uno tras el ángulo. Puede conseguirse un rectángulo sólido en el centro de la pantalla con **BOX 1,100,50,220,150,,1.** Observe que necesita una coma para indicar el valor del ángulo, incluso aunque no desee rotar la figura. Esto es así porque el ordenador lee la coma como si fuera el valor POR DEFECTO, es decir, que interpreta la falta de respuesta como una instrucción. Si no incluye Vd. la coma, el 1 del final de la línea se lee como el ángulo en que debe rotarse la figura.

Algunas formas típicas del comando **BOX** son:

Comando	**Efecto**
BOX on, columna 1, fila 1, columna 2, fila 2	Perímetro
BOX on, col 1, fila 1, col 2, fila 2, ángulo	Rotación
BOX on, col 1, fila 1, col 2, fila 2, lleno	Figura sólida
BOX off, col 1, fila 1, col 2, fila 2, ang, lleno	Borra un área de la pantalla

Columna 1, fila 1, son las posiciones en la pantalla de los vértices que Vd. especifique. Columna 1, fila 1 es el vértice superior izquierdo, mientras que columna 2, fila 2 es el vértice inferior derecho.

He aquí un programa que ilustra el funcionamiento del comando BOX (línea 60):

```
10 COLOR 0,1
20 COLOR 1,2
30 GRAPHIC 2,1
40 A=RND(1)*20+10
50 FOR L=0 TO 359 STEP A
60 BOX 1,100,30,220,130,L
70 NEXT L
80 FOR L=1 TO 2000:NEXT L
90 GRAPHIC 0,1
```

Dibujo de círculos

Su COMMODORE 16 tiene también un comando para dibujar círculos. Como en el comando **BOX**, puede Vd. variar la forma del círculo (para dibujar un óvalo o una elipse), y puede rotar dicho óvalo. Puede también dibujar solo una sección de la figura (un arco).

Las formas usuales del comando **CIRCLE** son:

Comando	**Efecto**
CIRCLE on, col.central, fila, centr, radio	círculo
CIRCLE on, col.c, fila, c, anchura, altura	óvalo
CIRCLE on, col.c, fil.c, anch, alt, inic, fin	arco
CIRCLE on, col.c, fil.c, anch, alt,,, ángulo	óvalo rotado
CIRCLE on, c.c,f.c, anch, alt,,,, ang.vert.	polígono

Este comando dibuja un círculo en el centro de la pantalla: **CIRCLE 1,160,100,50.** Esto le dice a su ordenador que debe dibujar un círculo en el centro en la columna 160, fila 100, con un radio de 50. De todos modos, puede que le salga un óvalo ya que los puntos de algunos televisores y monitores (los americanos, por ejemplo) son más altos que anchos. Para cambiar esto a un círculo real debe añadir un número separado para indicarle que la altura es diferente que la anchura. Por ejemplo: **CIRCLE 1,160,100,50,42.**

Dibujo de polígonos

Su COMMODORE 16 puede también dibujar un cuadrado, un triángulo o algún otro polígono utilizando el comando **CIRCLE.** Simplemente dígale al ordenador cuantos grados de círculo hay entre los vértices, de este modo: **CIRCLE 1,160,100,50,42,,,120.** Este comando dibuja un triángulo, ya que cada lado ocupa 120 grados. (Recuerde que la omisión de valores incluyendo comas en un comando gráfico causa que su ordenador tome los valores standard para ese dato). Una fórmula simple para obtener el ángulo de un polígono de N lados es 360/N.

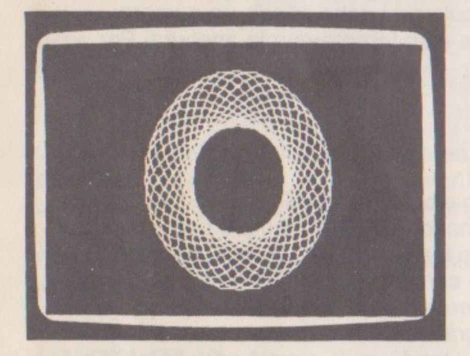
He aquí un rápido programa para dibujar polígonos:

```
10 GRAPHIC 2,1
20 INPUT"CUANTOS LADOS";A
30 IF A<2 OR A> 100 THEN PRINT "NO SEA
   RIDICULO":GOTO 20
40 CIRCLE 1,160,80,40,33,,,,360/A
50 GOTO 20
```

Puede Vd. elegir dibujar sólo un arco en vez de un círculo completo. El comando **CIRCLE** acepta el ángulo inicial y final, en grados, tras el valor de la altura. El comando **CIRCLE 1,160,100,50,42,90,180** representa solamente la parte inferior derecha del círculo. Para rotar un óvalo, añada el ángulo de rotación, en el sentido de las agujas del reloj, al final del comando, como en este ejemplo: **CIRCLE 1,160.100,100,20,,,30.**

He aquí un programa que utiliza el comando **CIRCLE** para conseguir un efecto interesante. No olvide limpiar la memoria del ordenador escribiendo NEW si tecleó el último programa.

```
10 COLOR 0,1
20 COLOR 1,2
30 GRAPHIC 1,1
40 A=RND(A)*20+10
50 FOR L=0 TO 359 STEP A
60 CIRCLE 1,160,100,80,40,,,L
70 NEXT L
80 FOR L= 1 TO 2000:NEXT L
90 GRAPHIC 0,1
```



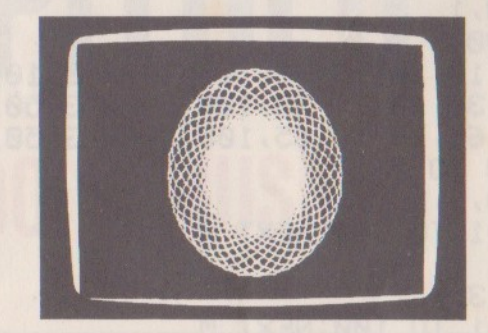
Este programa será ligeramente diferente cada vez que lo ejecute.

El comando PAINT

Para dibujar algo más que el perímetro de un círculo o de otra figura, puede utilizar el comando PAINT. Este comando llena cualquier área incluida en el interior de las líneas dibujadas en la pantalla. Si no hay ninguna línea dibujada, la pantalla se llena hasta los bordes. El comando BOX tiene una opción de llenado que colorea cuadrados y rectángulos. El comando PAINT puede colorear figuras irregulares y otras áreas no uniformes de la pantalla que no pueden ser rellenadas con otros comandos.

Para ver cómo funciona el comando PAINT, añada esta línea al último programa:


```
75 PAINT 1,160,100
```



GRAFICOS MULTICOLORES

Los gráficos de alta resolución del COMMODORE 16 le dan el control sobre cada punto o "pixel" de la pantalla, pero ya ha visto que la capacidad para colocar varios colores juntos se ve limitada. La mayor parte de programas hi-res pueden utilizar solamente uno o dos colores. Para incluir más colores diferentes, su ordenador tiene un modo gráfico especial llamado gráfico multicolor. En este modo, Vd. controla, en cada línea, la mitad de puntos que en el modo hi-res ya que cada uno de dichos puntos es el doble de ancho. Tiene Vd., por lo tanto, 160 puntos por línea y éstas siguen siendo 200. Hay una contrapartida en el uso de los colores múltiples, y es que se reduce ligeramente la resolución.

Para utilizar el modo gráfico multicolor, volvamos a repasar el comando **GRAPHIC,** visto anteriormente en este mismo capítulo. Verá que la pantalla multicolor sin texto se consigue con **GRAPHIC 3** y la pantalla multicolor con 5 líneas de texto, con **GRAPHIC 4.**

Ahora consulte la tabla del comando **COLOR.** Hay dos áreas que todavía no hemos usado, las áreas 2 y 3. Estas áreas le dan dos colores extra. Puede Vd. utilizar alguno de estos tres colores (1, el color del texto: 2, un color extra y 3, otro color extra). Estos colores no interfieren con ningún otro de la pantalla tal como lo hacían los colores en hi-res en algún programa anterior de este mismo capítulo.

Este programa hace uso de los gráficos multicolores, mostrando un efecto de "luz de neon".

```
10 COLOR 0,1
20 GRAPHIC 3,1
30 COLOR 3,1
40 TRAP 200
50 DRAW 3,10,10 TO 10,100:DRAW 3,10,55 TO 30,55
60 DRAW 3,30,10 TO 30,100:DRAW 3,50,10 TO 80,10
70 DRAW 3,65,10 TO 65,100:DRAW 3,50,100 TO 80,100
80 FOR L=0 TO 7
90 COLOR 3,2,L
100 FOR M=1 TO 100:NEXT M
110 NEXT L
120 COLOR 3,1
130 FOR M=1 TO 100:NEXT M
140 GOTO 80
200 GRAPHIC 0:COLOR 1,2,7
```

El área del color 3, la segunda de las áreas multicolores, tiene una peculiaridad que no poseen las otras. Una vez que haya dibujado en la pantalla usando el área 3, puede Vd. cambiar ese color en todos los lugares de la pantalla en los que aparezca, mediante el comando **COLOR**. Si activa Vd. el color con **COLOR 3,5** y dibuja utilizando ese color, sus gráficos aparecerán en púrpura. Si desea cambiar el color con **COLOR 3,6**, todas las áreas púrpura cambiarán al color verde. Esto no funciona con ninguna otra área. La Guía de Referencia del Programador —Serie 264— contiene más información sobre gráficos.

CAPITULO 7

SONIDO Y MUSICA

INTRODUCCION

He aquí un corto programa para hacer música en su COMMODORE 16. Tras teclearlo, cuando lo ejecute, aparecerá un interrogante en la pantalla. Teclee cualquier número desde 0 a 1015 como respuesta y pulse **RETURN.** Para detener el programa, introduzca el valor cero.

```
10 VOL 8
20 DO
30 INPUT X
35 IF X>1015 OR X<0 THEN PRINT"DE 0 A 1015,
   POR FAVOR":GOTO 30
40 SOUND 1,X,10
50 LOOP UNTIL X=0
```

Pulsando la tecla **RUN/STOP** también puede detener el programa.

He aquí como se toca una nota en su COMMODORE 16:

Primero: Escriba VOL 8 y pulse **RETURN.**

Segundo: Escriba SOUN 1,266,60 y pulse **RETURN.**

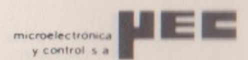
Oirá sonar una nota durante un segundo y, a continuación, se parará. Puede Vd. considerar esto como una sinfonía inacabada en el más completo sentido de la palabra (ha sonado una nota, faltan 3.500). Si no oye Vd. nada, suba el volumen de su monitor o su televisor e inténtelo de nuevo.

Estos dos pasos son los únicos comandos que necesita conocer para hacer música con su COMMODORE 16. Ambos son fáciles de comprender y aún más fáciles de usar.

El comando volumen

El comando **VOL** controla el VOLumen de las notas que interpreta su COMMODORE 16. El número que viene detrás de **VOL** es el valor del volumen. Este comando actúa de un modo muy parecido al del botón de volumen de su televisor. Cuando está puesto a cero **(VOL 0),** el volumen está desconectado y no puede Vd. oir nada. Cuando está a 8 **(VOL 8),** el volumen está al máximo y su ordenador suena tan fuerte como puede.

Intente de nuevo el primer ejemplo y utilice diferentes valores tras el comando **VOL.** Cuanto mayor sea el número, más fuerte sonará la nota.



El comando SOUND

El comando **SOUND** le dice a su ordenador todo lo que necesita saber sobre el sonido que Vd. desea obtener. El comando SOUND va seguido de tres números que describen la nota:

SOUN voz, valor de la nota, duración.

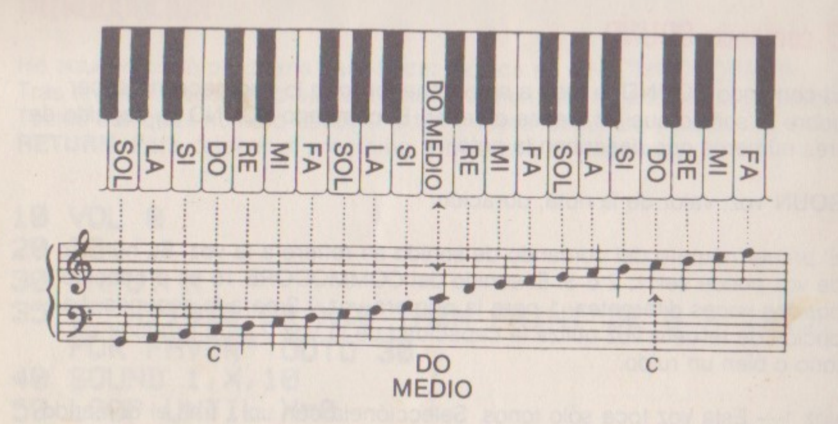
El primer número del comando de sonido se refiere a la voz. EL número de voz puede ser 1, 2 ó 3. El sonido del COMMODORE 16 es producido por dos voces diferentes, 1 para la primera voz y 2 para la segunda. La opción de tercera voz utiliza la capacidad de la voz 2 para producir un tono o bien un ruido.

Voz 1— Esta voz toca sólo tonos. Selecciónela con un 1 tras el comando SOUND.
Voz 2— Esta voz es como la voz 1, pero puede ser utilizada para tocar tonos o ruidos, Ponga un 2 en el comando para utilizar esta voz con tonos o un 3 para utilizarla con ruidos, para hacer efectos de sonido tales como truenos y lluvia.

El segundo número tras la palabra SOUND es el valor de la nota (frecuencia). Puede ser cualquier número entre 0 y 1015. Esto le dice a su COMMODORE 16 cuán grave o cuán aguda será la nota a tocar. Cuanto más altos sean los valores, más aguda será la nota. Los valores más altos (alrededor de 1015) son inaudibles para el oído humano.

Nota: Con la voz 3, el ruido es ''blanco'' solamente en el rango de 600 a 940. Puede utilizar valores fuera de estos límites para crear interesantes efectos de sonido.

Esto es una representación de las notas de la escala, con sus valores. Hay una tabla completa de notas para el COMMODORE 16 en los apéndices.



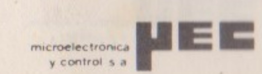
NOTA	LA	SI	DO	RE	MI	FA	SOL
VALOR	770	798	810	834	854	864	881
FRECUENCIA (HZ)	440.4	494.8	5227	588.7	658	699	782.2

Intente el siguiente programa:

```
10 VOL 8:REM ACTIVA EL VOLUMEN
20 X=0
30 DO
40 SOUND 1,X,5:REM  TOCA LA NOTA
50 X=X+5
60 LOOP UNTIL X=1015
70 VOL 0:REM DESCONECTA EL VOLUMEN
80 END
```

Este programa le muestra parte de la gama musical del COMMODORE 16.

El tercer número tras la palabra SOUND controla la duración (longitud) de la nota. Esto le dice a su ordenador durante cuanto tiempo debe tocar la nota. Este número puede ser cualquiera desde 0 a 65535. Este número activa un cronómetro que cuenta el tiempo en sesentavos de segundo. Una duración de 60, mantiene la nota durante un segundo. La regla es que cuanto mayor sea el número, durante más tiempo permanecerá sonando la nota. De hecho, si utiliza 65535, la nota durará aproximadamente unos 16 minutos. Para apagar un sonido, utilice una duración de 0, que no permite que se produzca ningún sonido.



Un efecto de sonido musical

Los efectos de sonido pueden crearse utilizando tonos musicales o bien ruidos. Combinando un programa simple en BASIC y comandos de sonido puede generar efectos poco usuales. Por ejemplo, el bucle FOR... NEXT... STEP puede utilizarse creativamente en efectos de sonido. Este comando inicia un bucle. Cada vez que el ordenador lee FOR, cambia el contador variable (S en este ejemplo). Cuando lee NEXT, vuelve a la instrucción FOR. Este programa utiliza un bucle FOR... NEXT con un valor negativo para STEP, que cuenta, en orden decreciente, desde el número mayor hasta el menor, de 25 en 25.

```
10 VOL 8:REM PONE EL VOLUMEN A 8
20 FOR S=1000 TO 700 STEP -25:REM INICIA
   UN BUCLE CON VALORES DESCENDENTES
30 SOUND 1,S,1
40 NEXT S
```

Escriba **RUN** y pulse **RETURN** para escuchar el efecto de sonido. La clave es la línea 20, que selecciona un número en el rango de 1000 a 700 bajando por la escala a intervalos de 25 números cada vez. Finalmente, la linea 30 ordena a su COMMODORE 16 que toque cada nota sólo un instante, poniendo la duración a 1, que significa 1/60 de segundo. El experimentar con diferentes números y valores de duración puede proporcionarle algunos efectos realmente interesantes.

CREACION DE EFECTOS DE SONIDO

Utilizar un valor de 3 cuando seleccione la voz en el comando SOUND, significa ruido. Esto se usa para crear efectos de sonido con ruido en vez de con un tono. El programa siguiente utiliza la voz 3 para crear el sonido de un vendaval.

```
10 VOL 2:REM CONTROLA EL NIVEL DE VOLUMEN
20 R=INT(RND(0)*10)+1:REM SELECCIONA UN
   NUMERO ALEATORIO DE 1 A 10
30 FOR X=1 TO R
40 SOUND 3,600+30*X,10
50 NEXT X
60 FOR X=R TO 1 STEP -1
70 SOUND 3,600+30*X,10
80 NEXT X
```




```
90 T=INT(RND(0)*100)+30
100 SOUND 3,600,T
110 GOTO 20
```

Las líneas 30 y 60 controlan los bucles FOR... NEXT que dan el valor de la nota (frecuencia), uno incrementándolo y el otro decrementándolo, en base al número aleatorio de la línea 20. Es importante que haya una variación en el tono ya que las ráfagas de viento de un vendaval varían en intensidad. Las líneas 40 y 70 son los comandos **SOUND** que crean el ruido. Las líneas 90 y 100 generan una leve demora aleatoria para simular la naturaleza desigual de un vendaval con lapsos de calma entre las ráfagas. El programa selecciona un número al azar que se utiliza para la duración de otro comando **SOUND.** Este comando se mantiene al mismo nivel y proporciona un fondo consistente de ruido que sirve como contrapunto a las ráfagas de viento. Esto puede parecer un poco complicado, como si necesitara ser un programador experimentado para ser capaz de hacerlo. Pero no es nada más que experimentación, intentar cosas diferentes y ver como se produce el ruido. La creación de efectos de sonido utilizando el ruido es un desafío, intentando captar los elementos correctos para producir el sonido que Vd. quiere exactamente. Para ser bueno en esto, debe estar dispuesto a experimentar.

Haciendo música

Quizás no comprenda todo lo que ocurre en estos programas, pero escríbalos, a pesar de todo, y vea lo que ocurre.

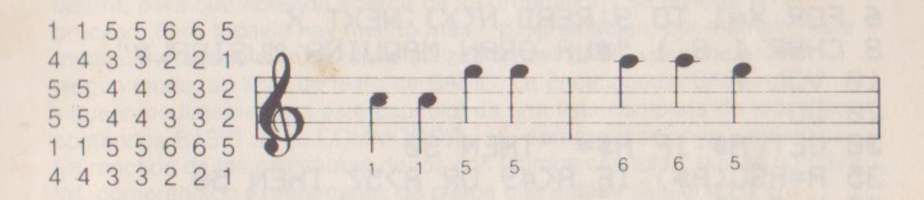
Este programa simula un piano utilizando las teclas del 1 al 8:

```
5 SCNCLR
10 FOR X=1 TO 8:READ N(X):NEXT X
20 VOL 8
30 DO
40 GET A$:IF A$=""THEN 40
50 A=ASC(A$):IF A<49 OR A>56 THEN 90
60 N=A-48
70 SOUND 1,N(N),5
80 COLOR 4,N,3
90 LOOP UNTIL A=32
100 VOL 0:COLOR 4,2,7
110 DATA 169,262,345,383,453,516,571,596
```


Pulse los números del 1 al 8 para tocar las notas. El borde de la pantalla incluso cambia de color con las diferentes notas. Cuando termine de tocar, pulse la barra espaciadora para detener el programa.

Ahora que su COMMODORE 16 puede ser usado como un piano, podría Vd. sentir la urgencia de tocar una canción popular. He aquí los números que debe pulsar para tocar una canción bien conocida que, seguramente, puede considerarse un clásico.

1 1 5 5 6 6 5
4 4 3 3 2 2 1
5 5 4 4 3 3 2
5 5 4 4 3 3 2
1 1 5 5 6 6 5
4 4 3 3 2 2 1



El siguiente programa interpreta una canción leyendo una lista de instrucciones **DATA.** Su ordenador lee los números contenidos en las instrucciones **DATA** como valores para asignar a variables (en este caso X e Y), cambiando los valores con cada bucle. Las instrucciones **DATA** van por parejas. El primer número es el valor de la nota y el segundo es el valor de la duración para el comando **SOUND.**

Row Boat

```
10 VOL 8
20 DO
30 READ X,Y
40 SOUND 1,X,Y
45 FOR D=1 TO Y*16+30:NEXT
50 LOOP UNTIL X=0:REM ESTE BUCLE CREA
   UNA BREVE PAUSA ENTRE LAS NOTAS
60 END
100 DATA 169,45,169,45,169,30
110 DATA 262,15,345,45,345,30
120 DATA 262,15,345,30,383,15
130 DATA 453,60,596,45,453,45
140 DATA 345,45,169,45,453,30
150 DATA 383,15,345,30,262,15
160 DATA 169,60
200 DATA 0,0
```

LA GRAN MAQUINA MUSICAL COMMODORE 16

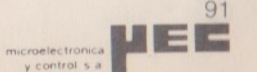
El último programa es un poco más largo. Es la "GRAN MAQUINA MUSICAL COMMODORE 16". Cuando pulse Vd. una tecla de 1 a 9, se tocará la nota y aparecerá dicha nota en el pentagrama, en la posición correcta.

```
5 GOSUB 1000
6 FOR X=1 TO 9:READ N(X):NEXT X
8 CHAR 1,8,1,"*LA GRAN MAQUINA MUSICAL*"
10 VOL 7
20 DO
30 GET A$:IF A$=""THEN 30
35 A=ASC(A$):IF A<49 OR A>57 THEN 50
36 N=A-48
40 SOUND 1,N(N),4
45 GSHAPE N$,150,8*(6+(9-N)),4
46 FOR Z=1 TO 50:NEXT Z
47 GSHAPE N$,150,8*(6+(9-N)),4
50 LOOP UNTIL A=32
55 VOL 0:GRAPHIC 0:SCNCLR
60 END
100 DATA 345,383,453,516,571,596,643,685,704
1000 GRAPHIC 1,1
1010 FOR Y=60 TO 124 STEP 16
1020 DRAW 1,100,Y TO 200,Y
1030 NEXT Y
1040 A$="FEDCBAGFE"
1050 FOR X=1 TO 9:C=13
1060 IF INT(X/2)=X/2 THEN C=14
1070 CHAR 1,C,X+6,MID$(A$,X,1),0
1075 CHAR 1,C+10,X+6,RIGHT$(STR$(10-X),1)
1080 NEXT X
1090 FOR X=1 TO 8:FOR Y=11 TO 16:DRAW
   1,X,Y:NEXT Y,X
1100 Y=1:X=8:DRAW 1,8,16 TO X,Y
1110 SSHAPE N$,1,1,8,16
1120 GSHAPE N$,1,1,4
1130 RETURN
```




Como puede ver, la música y el sonido pueden utilizarse para realzar programas o ser el foco mismo del programa. Los ejemplos de este capítulo solamente le pueden dar una idea de la capacidad musical de su COMMODORE 16. No tenga miedo de intentar nuevos sonidos y ruidos ni de crear su propia obra maestra.

Por ahora, ha empezado Vd. a manejar algunas de las cosas que puede hacer con su COMMODORE 16. El objetivo de este manual ha sido darle una idea del ordenador, para que Vd. intente cosas nuevas y vea lo que ocurre, para que aprenda acerca de su ordenador y se divierta en el proceso. Pero todavía hay mucho más... programación, por ejemplo. Este manual ha tocado el inicio de la utilización del BASIC, pero no es, de ningún modo, un libro de texto de BASIC. La Enciclopedia BASIC (que sigue inmediatamente a este capítulo) da una lista completa de todos los comandos BASIC de su COMMODORE 16, con explicaciones y ejemplos. En muchos de los programas de los dos últimos capítulos, quizás no haya Vd. comprendido exactamente los pasos que estaba siguiendo. Para aprender más sobre programación en lenguaje BASIC, debería adquirir libros preparados para enseñar BASIC. Una lista de esos libros aparece en el apéndice. Si está Vd. realmente interesado en programación, puede desear adquirir la Guía de Referencia del Programador —Serie 264—, que está enfocada al como y el por qué de la programación, revelando los trucos secretos (y no tan secretos) del tema. Ahora que ha terminado de leer este manual, puede pensar que se le ha abandonado a su suerte, pero no es así exactamente. Puede Vd. leer cualquiera de las muchas revistas sobre ordenadores, algunas de las cuales están dedicadas por entero a los ordenadores COMMODORE, periféricos y software. También puede Vd. formar parte de algún grupo de gente próxima a Vd. que comparta su interés por los ordenadores COMMODORE, UN CLUB DE USUARIOS COMMODORE. Para más información sobre esos grupos, contacte con el Centro de Información CBM (Microelectrónica y Control).



BASIC 3.5

ENCICLOPEDIA

- Introducción
- Comandos BASIC
- Instrucciones BASIC
- Funciones
- Variables y operadores
- Abreviaturas BASIC y tabla de referencias

INTRODUCCION

Ha encontrado Vd. en este manual una serie de ejercicios utilizando el lenguaje BASIC, que le habrán dado una idea de la programación y del vocabulario empleado. Esta enciclopedia le ofrece una lista completa de las reglas y términos (SINTAXIS) del lenguaje BASIC 3.5, junto con una concisa descripción de cada uno. Experimente con estos comandos y recuerde que no puede Vd. estropear su ordenador escribiendo programas y que el mejor método para aprender programación es haciéndola.

La enciclopedia le facilita los formatos y unas breves explicaciones con ejemplos de las instrucciones y comandos del BASIC 3.5. No se trata de enseñarle BASIC. Si está Vd. interesado en aprenderlo, en el apéndice G encontrará una lista de libros que le ayudarán.

Los comandos e instrucciones están relacionados en secciones separadas. Dentro de las secciones, los comandos e instrucciones van listados por orden alfabético. Los comandos se utilizan, principalmente, en modo directo, mientras que las instrucciones se usan más frecuentemente en programas. En muchos casos, los comandos pueden ser utilizados, como las instrucciones, en un programa si les antepone un número de línea. Puede utilizar muchas instrucciones como comandos usándolas en modo directo (por ejemplo, sin número de línea). Si está inseguro acerca de donde estará colocado un término, están todos ellos listados en la tabla de referencia.

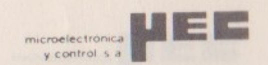
La Enciclopedia BASIC está organizada del modo siguiente:

- **COMANDOS:** Los comandos usados para trabajar con programas, corregirlos, almacenarlos y borrarlos.
- **INSTRUCCIONES:** Las instrucciones de programas BASIC usadas en las líneas numeradas del programa.
- **FUNCIONES:** Las funciones de cadena, numéricas y de escritura.
- **VARIABLES Y OPERADORES:** Los diferentes tipos de variables, nombres legales de variables, y operadores lógicos y aritméticos.

Una explicación más completa de los comandos BASIC 3.5 la encontrará en la Guía de Referencia del Programador —Serie 264—, disponible en su distribuidor COMMODORE.

FORMATO DE LOS COMANDOS E INSTRUCCIONES

Los comandos e instrucciones presentados en esta sección de la enciclopedia siguen unas convenciones de formato para hacerlos lo más claros posible. En muchos casos, hay varios ejemplos para ilustrar el aspecto del



comando. El ejemplo siguiente muestra algunas de las convenciones de formato utilizadas en los comandos e instrucciones BASIC:

EJEMPLO: **LOAD** ''nombre del programa'', **DO, U8**

Palabra clave — argumento — argumentos adicionales (posiblemente opcionales)

Las partes del comando o instrucción que debe Vd. teclear exactamente tal como aparecen van subrayadas, mientras que el nombre del comando va en mayúsculas. Las palabras que no necesita teclear exactamente, tal como el nombre del programa, van impresas en letra normal. Cuando aparecen comillas ('' ''), usualmente incluyendo el nombre de un programa o de un fichero, debe incluirlas en el comando o instrucción, tal como en el ejemplo.

- Las PALABRAS CLAVE aparecen en mayúsculas y en negrita. DEBE VD. INTRODUCIR ESTAS PALABRAS EXACTAMENTE TAL COMO APARECEN. Por otro lado, muchas palabras clave tienen abreviaturas que puede Vd. también utilizar (vea la tabla de referencia).

Las palabras clave son palabras que forman parte del lenguaje BASIC que su ordenador conoce. Las palabras clave son la base de la instrucción o el comando. Le indican al ordenador que tipo de acción quiere Vd. realizar. Estas palabras no pueden ser utilizadas como nombres de variable.

- Los ARGUMENTOS (también llamados parámetros) aparecen en minúscula. Los argumentos son la parte de un comando o instrucción que selecciona Vd.. Completan la palabra clave dándole la información específica sobre el comando o instrucción. Por ejemplo, una palabra clave le dice al ordenador que cargue un programa, mientras que el argumento le dice qué programa específico debe cargar y el segundo argumento le especifica en qué unidad de disco está contenido el programa. Los argumentos incluyen nombre de ficheros, variables, números de línea, etc.

- Los CORCHETES ([ ]) indican argumentos opcionales. Vd. selecciona alguno o ninguno de los argumentos listados, según sus requerimientos.

- Los PARENTESIS ANGULARES < > indican que DEBE Vd. elegir uno de los argumentos listados.

- La BARRA VERTICAL . separa elementos en una lista de argumentos en la que su elección está limitada a uno de ellos y no puede Vd. utilizar ningún otro. Cuando la barra vertical aparece en una lista encerrada entre CORCHETES, su elección está limitada a los elementos de la lista, pero tiene la opción de no utilizar ninguno de ellos.

- PUNTOS SUSPENSIVOS... Una secuencia de tres puntos, significa que una opción o argumento puede ser repetido más de una vez.



• LAS COMILLAS " " encierran cadenas de caracteres, nombres de ficheros y otras expresiones. Cuando los argumentos van incluidos entre comillas, debe Vd. incluir las comillas en su comando o instrucción. Las comillas son parte necesaria de un comando o instrucción.

• PARENTESIS ( ) Cuando los argumentos van entre paréntesis, debe Vd. incluir los paréntesis en el comando o instrucción. Los paréntesis también son necesarios cuando aparecen en la descripción de un comando o instrucción.

• VARIABLE se refiere a cualquier nombre válido de variable BASIC, tal como X, A$ o T%.

• EXPRESION significa cualquier expresión BASIC válida, tal como A+B+2 o 5*(x+3).

COMANDOS BASIC

AUTO

AUTO [línea #]

Activa la numeración automática de líneas que facilita la introducción de programas escribiendo en su lugar los números de línea. Al introducir cada línea de programa y pulsar **RETURN,** el siguiente número de linea se imprime en la pantalla, con el cursor en la posición de inicio de dicha línea. El argumento [línea #] se refiere al incremento entre números. AUTO SIN ARGUMENTO desconecta la numeración automática, al igual que RUN. Esta instrucción es ejecutable solamente en modo directo.

EJEMPLOS:

AUTO 10	numera las líneas automáticamente en incrementos de 10
AUTO 50	numera las líneas automáticamente en incrementos de 50
AUTO	desconecta la numeración automática

BACKUP

BACKUP Ddrive # TO Ddrive # [,ON Uunidad #]

Este comando copia todos los ficheros de un disckette a otro en una unidad de discos doble. Puede Vd. copiar a un nuevo diskette sin utilizar previamente el comando HEADER para formatearlo ya que el comando BACKUP copia toda la información del diskette, incluyendo el formato. Debería Vd. siempre hacer BACKUP de los diskettes importantes para el caso de que el original se perdiera o se estropeara.

Ya que el comando BACKUP también formatea los diskettes, destruye cualquier información previa en el diskette en el que se está haciendo la copia. Por lo tanto, si está Vd. utilizando para la salida un diskette previamente utilizado, asegúrese de que no contiene programas que desee conservar. Vea también el comando COPY.

> **NOTA:** Este comando sólo puede ser utilizado con unidades de disco dobles.

EJEMPLOS:

BACKUP D0 TO D1	Copia todos los ficheros del disco el DRIVE 0 al disco en el DRIVE 1
BACKUP D0TOD1, ON U9	Copia todos los ficheros del DRIVE 0 al DRIVE 1 en la unidad de discos 9

COLLECT

COLLECT [D drive #] [,ON Unidad #]

Utilice este comando para liberar el espacio asignado a ficheros mal cerrados y borrar del directorio toda referencia a estos ficheros.

EJEMPLO:

COLLET D0

CONT

CONT (Continuar)

Este comando se utiliza para relanzar un programa tras haberlo parado utilizando la tecla STOP, la instrucción STOP o una instrucción END en el programa. El programa se continuará ejecutando en el punto en que se detuvo. CONT no funcionará si ha cambiado o añadido líneas del programa (o incluso, si ha movido el cursor a una línea del programa y ha pulsado **RETURN** sin cambiar nada en ella), si el programa se ha detenido a causa de un error o si ha causado Vd. un error antes de intentar relanzar el programa. En ese caso el error que le aparecerá sera CAN'T CONTINUE ERROR.

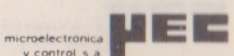
COPY

COPY [D drive # ,] ''fich. origen'' **TO** [D drive # ,] ''otro fich.'' [,ON Uunidad #]

Copia un fichero del disco en un DRIVE (el fichero de origen) al disco en el otro DRIVE, solamente en una unidad doble, o crea una copia de fichero en el mismo DRIVE (con un nombre de fichero diferente).

EJEMPLOS:

COPY D0, "MEDIODIA" TO D1, "NOCHE"	Copia MEDIODIDA del DRIVE 0 al DRIVE 1, dejándolo como NOCHE
COPY D0, "BASURA" TO D1, "BASURA"	Copia BASURA del DRIVE 0 al 1
COPY DO TO D1	Copia todos los ficheros del DRIVE 0 al DRIVE 1
COPY "GATOS" TO "PERROS"	Copia GATOS como un programa llamado PERROS en el mismo DRIVE



DELETE

DELETE [Primera línea #] [-última línea #]

Borra líneas de texto en BASIC. Este comando sólo se puede ejecutar en modo directo.

EJEMPLOS:

DELETE 75	Borra la línea 75
DELETE 10-50	Borra desde la línea 10 a la 50, ambas inclusive.
DELETE-50	Borra todas las líneas desde el inicio del programa hasta la línea 50 inclusive.
DELETE 75–	Borra todas las líneas desde la 75 hasta el final del programa.

DIRECTORY

DIRECTORY [D drive #] [,Uunidad #] [''nombre del fichero'']

Lista el directorio de un disco en la pantalla del COMMODORE 16. Utilice **CTRL-S** para detener el listado (cualquier otra tecla relanza el listado tras la pausa). Utilice la tecla C= (la tecla COMMODORE) para ralentizarlo. El comando DIRECTORY no se puede utilizar para imprimir una copia. Para hacer esto, debe Vd. cargar el directorio del disco (destruyendo el programa actualmente en memoria).

EJEMPLOS:

DIRECTORY	Lista todos los ficheros de un disco.
DIRECTORY D1, U9, "TRABAJO"	Lista el fichero de la unidad 9 (8 por defecto), DRIVE 1, llamado TRABAJO.
DIRECTORY "AB*"	Lista todos los ficheros que empiecen cpn las letras ''AB'' como ABECEDARIO, ABEJA, etc.

DIRECTORY D0, "FICHERO?.BAK" El signo ? es un comodín que sustituye a cualquier carácter que ocupe esa posición: FICHERO1.BAK, FICHERO2.BAK, FICHERO3.BAK son todos iguales a esa cadena.

NOTA: Para imprimir el directorio del DRIVE 0, unidad 8, utilice lo siguiente:

```
LOAD "$0", 8
OPEN4,4:CMD4:LIST
PRINT # 4:CLOSE 4
```

DLOAD

DLOAD "nombre del fichero" [,Ddrive #] [,Uunidad #]

Este comando carga un programa desde el disco a la memoria. (Utilice LOAD para cargar programas desde cinta). Debe Vd. indicar un nombre de programa.

EJEMPLOS:

DLOAD "DCAMION"	Busca en el disco el programa "DCAMION" y lo carga.
DLOAD (A$)	Carga un programa desde disco cuyo nombre está en la variable A$. Recibirá un mensaje de error si A$ está vacía.

El comando DLOAD puede ser utilizado en un programa en BASIC para encontrar y lanzar otro programa desde el disco. Esto se llama encadenar programas.

DSAVE

DSAVE "nombre del fichero" [,D drive #] [,Uunidad #]

Este comando almacena un programa en disco. (Utilice SAVE para almacenar programas en cinta). Debe especificar un nombre de programa.

EJEMPLOS:

DSAVE "DIA D"	Salva a disco el programa "DIA D"
DSAVE (A$)	Salva a disco el programa cuyo nombre se encuentra en la variable A$
DSAVE "PROG3",D0,U9	Salva el programa "PROG3" a la unidad de disco número 9.

HEADER

HEADER "nombre del disco", D drive # [,I id. #] [,ON Uunidad #]

Antes de utilizar un diskette por primera vez debe formatearlo con el comando HEADER. Si desea borrar completamente un diskette para volverlo a utilizar puede también utilizar este comando. Este comando divide el disco en secciones llamadas bloques y crea una tabla del contenido, llamada directorio o catálogo del disco. El nombre del disco puede ser cualquier nombre de hasta 16 caracteres. El número de identificación está formado por dos caracteres cualesquiera. Dé a cada disco un único número de identificación. Tenga cuidado al utilizar el comando HEADER ya que borra todos los datos almacenados en el disco. Si no da número de identificación realizará un formateado más rápido. Se utilizará el número anterior. Sólo puede utilizar el método rápido si el disco ha sido ya formateado previamente, ya que el formateado rápido sólo borra el directorio en vez de formatear el disco.

EJEMPLOS:

HEADER "MYDISK",I23,DO

HEADER "LAPELOTA",I45,D1,U8

HELP

HELP

El comando HELP se utiliza tras haber recibido un mensaje de error en su programa. Cuando escribe HELP, la línea en la que ha ocurrido el error aparece en la pantalla con la porción errónea representada en caracteres destellantes.



KEY

KEY [tecla # , cadena]

Hay ocho (8) teclas de función disponibles para el usuario en su COMMODORE 16. Cuatro de ellas al pulsar **SHIFT** y cuatro sin pulsar **SHIFT.** Su COMMODORE 16 le permite definir que hará cada una de estas teclas cuando sea pulsada. Key sin ningún parámetro proporciona un listado de las funciones asignadas en ese momento. Los datos que asigna a una tecla se escriben tras pulsar la tecla. La longitud máxima para todas las definiciones es, en conjunto, de 128 caracteres. Pueden ser asignados a una tecla comandos o series de comandos. Por ejemplo:

KEY 7, "GRAPHIC0" + CHR$(13) + "LIST" + CHR$(13)

hace que el ordenador seleccione el modo texto y liste el programa cada vez que la tecla "F7" sea pulsada (en modo directo). CHR$(13) es el código ASCII de **RETURN.** Utilice CHR$(34) para incorporar dobles comillas a una cadena.

Las teclas pueden ser redefinidas en un programa. Por ejemplo:

```
10 KEY 2, "TESTING" + CHRZ$(34):KEY 3, "NO"
10 FOR I = 1 TO 8: KEY I,CHR$(I + 132):NEXT
```

define las teclas de función tal como están definidas en el COMMODORE 64 y VIC-20.

Para restaurar todas las teclas de función a sus valores por defecto, restaure su COMMODORE 16 desconectándolo y volviéndolo a conectar, o pulse el botón de RESET.



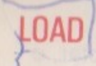
LIST

LIST [Primera línea] [-[última línea]]

El Comando LIST le permite ver las líneas de un programa BASIC que haya Vd. escrito o cargado en la memoria del COMMODORE 16. Cuando se utiliza solamente LIST (sin ningún número a continuación), se obtiene un listado del programa por la pantalla, que puede ser ralentizado pulsando la tecla C=, detenido con CTRL-S (reiniciado con cualquier otra tecla), o parado pulsando la tecla RUN/STOP. Si añade a la palabra LIST un número de línea, su COMMODORE 16 solamente listará esa línea. Si escribe Vd. LIST seguido por dos números separados por un guión, el COMMODORE 16 le mostrará todas las líneas desde el primer número hasta el segundo. Si escribe LIST seguido por un número y un guión, le mostrará todas las líneas desde ese número hasta el final del programa. Si Vd. escribe LIST, un guión y, después, un número tendrá el listado de todas las líneas desde el principio del programa hasta ese número. Utilizando estas variaciones, puede Vd. examinar cada porción del programa o representar fácilmente cualquier línea en la pantalla para modificarla.

EJEMPLOS:

LIST	Muestra el programa completo
LIST 100-	Muestra desde la línea 100 hasta el final.
LIST 10	Muestra solamente la línea 10.
LIST -100	Muestra desde el principio hasta la línea 100.
LIST 10-200	Muestra desde la línea 10 hasta la 200, ambas inclusive.



LOAD [''nombre del fichero'' [, Periférico #] [,dirección]]

Este es el comando que deberá usar cuando quiera utilizar programas almacenados en cinta cassette o disco. Si escribe simplemente LOAD y pulsa **RETURN,** la pantalla del COMMODORE 16 se queda en blanco. Pulse el botón de Play y el COMMODORE 16 empezará a buscar un programa en la cinta. Cuando encuentre uno, el ordenador imprimirá FOUND ''nombre del fichero''. Puede Vd. pulsar la tecla C= para cargarlo. Si no lo hace, el ordenador continúa buscando tras un breve intervalo. Una vez que el programa está cargado, puede Vd. ejecutarlo, listarlo o modificarlo.

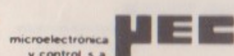
También puede escribir la palabra LOAD seguida por el nombre de un programa, que habitualmente será un nombre entre comillas (''nombre del programa''). El nombre puede ir seguido por una coma (fuera de las comillas) y un número (o una variable numérica), que actúa como número de periférico para determinar donde está almacenado el programa (disco o cinta). Si no se dá ese número, el ordenador asume el periférico 1, que es el grabador de cinta.

El otro periférico utilizado comúnmente con el comando LOAD es la unidad de disco, que es el periférico número 8.

EJEMPLOS:

LOAD	Lee el primer programa de cinta.
LOAD "BASES"	Busca en la cinta el programa llamado ''BASES'' y lo carga al encontrarlo.
LOAD A$	Busca un programa cuyo nombre se encuentra en la variable A$
LOAD "PUNTA",8	Busca el programa llamado PUNTA en la unidad de disco y lo carga cuando lo encuentra.

El comando LOAD puede ser utilizado en un programa BASIC para encontrar y ejecutar el siguiente programa de una cinta. Esto se llama encadenar programas.



La DIRECCION determina el lugar de la memoria en que va a cargar el programa. Una dirección de 0 le indica al COMMODORE 16 que cargue el programa al inicio del area de programas BASIC y una dirección de 1 le indica que debe cargarlo en el lugar desde el que se salvó. El valor por defecto de la dirección es 0. Esto se utiliza generalmente en caso de cargar programas en código máquina.



NEW

Este comando borra enteramente el programa en memoria y limpia todas las variables que hayan podido ser utilizadas. A menos que el programa haya sido almacenado en algún sitio, se pierde hasta que lo vuelva a escribir. Tenga cuidado al utilizar este comando.

El comando NEW puede también ser utilizado como una instrucción en un programa BASIC. Cuando su COMMODORE 16 llegue a esta línea, el programa es borrado y se detiene todo. No es muy utilizado en circunstancias normales.

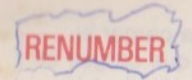
RENAME

RENAME [D drive # ,] ''nombre viejo'' TO ''nombre nuevo'' [,Uunidad #]

Se utiliza para dar un nuevo nombre a un fichero en diskette.

EJEMPLOS

RENAME D0, "ACTIVO"TO"PASIVO"	Cambia el nombre del fichero de ACTIVO a PASIVO



RENUMBER [nueva línea de inicio # [, línea de inicio vieja #]]]

La nueva línea de inicio es el número de la primera línea del programa tras la renumeración. Por defecto, se toma el valor 10.

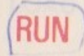
El incremento es el espacio entre los números, por ejemplo: 10, 20, 30, etc. También es 10, por defecto.

La línea de inicio vieja es el número de línea en que hay que empezar a renumerar. Esto le permite renumerar una sección de programa. Por defecto, se toma la primera línea del programa.

Este comando sólo puede ejecutarse en modo directo.

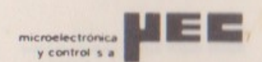
EJEMPLOS:

RENUMBER 20,20,1	Empezando en la línea 1, renumera el programa. La línea 1 pasa a ser la 20 y las restantes líneas son numeradas con un incremento de 20.
RENUMBER,,65	Empezando en la línea 65, renumera con un incremento de 10. La línea 65 pasa a ser la 10 (a menos que ya existan las líneas 10-64, en cuyo caso, el comando no se leva a efecto.



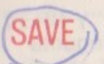
RUN [línea #]

Una vez que el programa ha sido tecleado o cargado en la memoria, el comando RUN hace que empiece a trabajar. RUN limpia todas las variables del programa antes de empezar la ejecución. Si no hay ningún número a continuación del comando RUN, el ordenador empieza por la línea del programa con número menor. Si hay un número a continuación del comando, la ejecución empieza en esa línea. RUN puede ser usado dentro de un programa.



EJEMPLOS:

RUN	Empieza el programa desde la línea con el número más bajo.
RUN 100	Empieza el programa en la línea 100.



SAVE [''nombre del fichero'' [,Periférico # [,EOT]]]

Este comando almacena el programa actualmente en la memoria en una cinta de cassette o en un disco. Si escribe simplemente la palabra SAVE y pulsa RETURN, su COMMODORE 16 almacena el programa en una cinta de cassette. No hay manera de comprobar si ya hay un programa en ese lugar de la cinta, por lo que tenga cuidado con ellas. Si escribe Vd. el comando SAVE seguido de un nombre entre comillas o el nombre de una variable de cadena, el COMMODORE 16 le da ese nombre al programa, de ese modo puede ser más fácil localizarlo y recuperarlo en el futuro. Si quiere especificar un número de periférico, ponga detrás del nombre una coma (tras las comillas) y un número o una variable numérica. El periférico número 1 es la unidad de cintas y el número 8 es el disco. Tras el número, en el caso de las cintas, puede haber una coma y un segundo número, entre 0 y 3. Si este segundo número es 2, el COMMODORE 16 pone un marcador de FIN DE CINTA (EOT) tras el programa. Si intenta cargar un programa y el COMMODORE 16 encuentra uno de estos marcadores en vez del programa, le dará el mensaje FILE NOT FOUND ERROR.

EJEMPLOS:

SAVE	Almacena el programa en cinta, sin nombre.
SAVE "DINERO"	Almacena en cinta bajo el nombre ''DINERO''.
SAVE A$	Almacena en cinta con el nombre que se encuentra en la variable A$.
SAVE"VD",8	Almacena en disco bajo el nombre VD.

SAVE"J",1,2 — Almacena en cinta bajo el nombre J y coloca un marcador de FIN DE CINTA tras el programa.

SCRATCH

SCRATCH "nombre del fichero" [D drive #] [,UUnidad #]

Borra un fichero del directorio de un disco. Como precaución, le pregunta "Are you sure?" (¿Está Vd. seguro?) antes de que el COMMODORE 16 complete la operación. Pulse Y para efectuar el borrado o N para cancelar la operación. Utilice este comando para borrar ficheros inservibles, para crear más espacio en el disco.

EJEMPLO:

SCRATCH"MYBACK",D1 — Borra el fichero MYBACK del disco en el drive 1.



VERIFY

VERIFY "nombre del fichero" [,Periférico #] [,dirección]

Este comando hace que su COMMODORE 16 compruebe el programa en cinta o disco con el que tiene en memoria. Esto es para probar si el programa que acaba de salvar está realmente salvado, asegurándose de que no ha habido ningún problema. Este comando también es muy útil para posicionar una cinta de manera que su ordenador empiece a escribir a continuación del último programa grabado en la cinta. Todo lo que debe hacer es decirle al COMMODORE 16 que verifique el último programa de la cinta. Lo hará y le dirá que los programas no coinciden (lo que Vd. ya sabía). Ahora la cinta está donde Vd. quería y puede almacenar el siguiente programa sin miedo a borrar uno anterior.

VERIFY sin nada más tras el comando tiene por efecto que el COMMODORE 16 compruebe el primer programa de la cinta, sea cual sea su nombre, con el que guarda en la memoria. VERIFY seguido por el nombre de un programa (entre comillas) o una variable de cadena, busca dicho programa en la cinta y, a continuación, lo comprueba. VERIFY seguido por un nombre, una coma y un número comprueba el programa

en la unidad de dicho número (1 para la cinta, 8 para el disco). La dirección tiene la misma utilidad que en el comando LOAD.

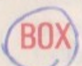
EJEMPLO:

VERIFY — Comprueba el primer programa de la cinta

VERIFY"REAL" — Busca REAL en la cinta y lo comprueba con la memoria,

VERIFY"YO",8,1 — Busca yo en el disco y, a continuación, lo comprueba.

INSTRUCCIONES BASIC



BOX

BOX [color #], a1, b1, [a2, b2] [,ángulo] [,lleno]

color	Origen del color (0-3). Por defecto es 1 (color de los caracteres)
a1, b1	Coordenadas del vértice
a2, b2	Vértice opuesto a a1, b1. Por defecto es el PC.
ángulo	Rotación en grados. Por defecto es 0 grados.
lleno	Rellenar la figura de color (O: no, 1: si). Por defecto es 0.

Este comando le permite dibujar un rectángulo de cualquier tamaño en cualquier lugar de la pantalla. Para obtener el valor por defecto, incluya una coma sin especificar valor. La rotación es en relación al centro del rectángulo. El Cursor de Pixels (PC), queda en la posición a2,b2 tras ejecutar la instrucción BOX.

EJEMPLOS:

BOX 1, 10, 10, 60, 60 — Dibuja el perímetro de un rectángulo.

BOX, 10, 10, 60, 60, 45, 1 — Dibuja un cuadrado sólido rotado (un rombo).

BOX, 30, 90, , 45, 1 — Dibuja un polígono sólido rotado.



CHAR [color #], x, y, ''cadena'' [,inverso]

color	Origen del color (0-3)
x	Columna del carácter (0-39)
y	Fila del carácter (0-24)
''cadena''	Texto a imprimir
inverso	Indicador de campo inverso (O = no, 1 = si)

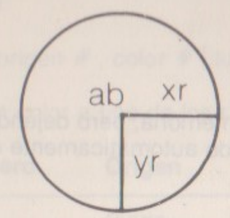
El texto (una cadena alfanumérica) puede ser representado con cualquier tipo de pantalla, en una posición determinada, con el comando CHAR. Los datos de los caracteres se leen del área de caracteres ROM del COMMODORE 16. Debe Vd. indicar las coordenadas del primer carácter y la cadena de texto que quiere representar. El origen del color y la imágen son opcionales.

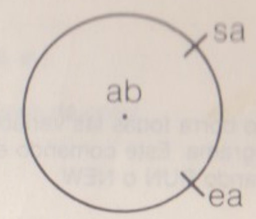
La cadena contínúa en la línea siguiente si sobrepasa el borde derecho de la pantalla. Cuando se utiliza en modo TEXTO, la cadena impresa por el comando CHAR actúa, simplemente, como una instrucción PRINT, incluyendo campo inverso, cursores, parpadeo on/off, etc. Estas funciones de control dentro de la cadena no funcionan cuando el comando CHAR es utilizado para representar un texto en modo GRAFICO.

CIRCLE

CIRCLE [c], [a,b], rx[, [ry] [, [ia] [, [fa] [, [ángulo] [,inc]]]]]

c	Origen del color (O-3)
a,b	Coordenadas del centro (por defecto las del Cursor de Pixels [PC])
rx	Radio x
ry	Radio y (por defecto, rx)
ia	Angulo de inicio del arco (por defecto, 0)
fa	Angulo de final del arco (por defecto, 360)
ángulo	Rotación en grados (por defecto, 0 grados)
inc	Grados entre los segmentos (por defecto 2 grados)





Con el comando CIRCLE puede Vd. dibujar un círculo, una elipse, un arco, un triángulo o un octógono. La coordenada final está en la circunferencia del círculo al final del ángulo de arco. Cualquier rotación es sobre el centro. Los arcos son dibujados empezando por el ángulo final, en el sentido de las agujas del reloj. El incremento de segmentos controla la perfección de la figura. Cuanto más bajos sean los valores de inc, las figuras serán más redondeadas.

EJEMPLOS:

CIRCLE, 160,100,65,10	Dibuja una elipse
CIRCLE, 160,100,65,50	Dibuja un óvalo
CIRCLE, 60,40,20,18,,,,45	Dibuja un octógono
CIRCLE, 260,40,20,,,,,90	Dibuja un rombo
CIRCLE, 60,140,20,18,,,,120	Dibuja un triángulo

CLOSE

CLOSE fichero #

Este comando completa y cierra cualquier fichero abierto por una instrucción OPEN. El número que sigue a la instrucción CLOSE es el número del fichero que debe ser cerrado.

EJEMPLO:

CLOSE 2	Se cierra el fichero lógico 2.



CLR

Este comando borra todas las variables de la memoria, pero dejando intacto el programa. Este comando es ejecutado automáticamente cuando se da el comando RUN o NEW.

CMD

CMD fichero # [,cadena]

CMD envía la salida que normalmente se realiza por la pantalla (por ejemplo, instrucciones PRINT, LIST, pero no POKES en la pantalla) a otro periférico. Este puede ser una impresora, o un fichero de datos en cinta o disco. Este periférico o fichero debe ser previamente abierto. El comando CMD debe ir seguido por un número o una variable numérica refiriéndose al fichero.

EJEMPLOS:

OPEN 1,4	Abre el periférico # 4, que es la impresora.
CMD 1	Todas las salidas normales van ahora a la impresora.
LIST	El listado sale por la impresora en vez de la pantalla. Incluso la palabra READY.
PRINT 1	Devuelve la salida a la pantalla.
CLOSE 1	Cierra el fichero.



COLOR origen # , color # [,luminosidad #]

Asigna un color a uno de los cinco orígenes de color:

Número	Origen
0	fondo
1	caracteres
2	multicolor 1
3	multicolor 2
4	borde

Los colores que puede Vd. utilizar, están en el rango de 1 a 16 (1 es el negro, 2 es el blanco, 9 es naranja, etc. según las teclas de color). Como opción, puede Vd. incluir el nivel de luminosidad de 0 a 7, siendo 0 el menor y el 7 el mayor. Por defecto este valor es 7. La luminosidad le permite seleccionar ocho niveles de brillantez para cada color, excepto el negro.



DATA lista de constantes separadas por comas

Esta instrucción va seguida de una lista de elementos que serán usados por una instrucción READ. Los elementos pueden ser números o palabras y deben ir separados por comas. Las palabras no necesitan ir entre comillas a menos que contengan uno de los siguientes caracteres: ESPACIO, dos puntos o coma. Si se ponen dos comas sin nada entre ellas, el valor será leído como un cero, si es un número, o como una cadena vacía. La instrucción DATA debe formar parte de un programa, de otro modo no será aceptada. Vea también la instrucción RESTORE, que permite a su COMMODORE 16 releer datos.

EJEMPLO:

DATA 100, 200, FRED "WILMA",, 3.14,ABC123

DEF FN

DEF FN nombre (variable) = expresión

Este comando le permite definir un cálculo complejo como una función. En el caso de que una fórmula larga deba ser utilizada varias veces en un programa, puede ahorrar bastante espacio.

El nombre que Vd. le dé a la función numérica empieza con las letras FN, seguidas por cualquier nombre legal de variable numérica. Primero debe Vd. definir la función utilizando la instrucción DEF, seguida por el nombre que le de Vd. a la función. A continuación del nombre, hay un par de paréntesis ( ) con una variable numérica incluida en ellos (en este caso, X). Luego viene un signo igual, seguido por la fórmula que desea definir. Puede Vd. ''llamar'' a la fórmula, sustituyendo cualquier número por X, utilizando el formato mostrado en la línea 20 del ejemplo siguiente:

EJEMPLO:

```
10 DEF FNA(X)=12*(34.75-X/3)+X

20 PRINT FNA(7)
```

El número 7 es insertado en cada lugar en que aparezca una X en la fórmula dada en la instrucción DEF.

NOTA: DEF FN Puede solamente utilizarse con funciones numéricas standard, no con funciones enteras ni de escritura.



DIM

DIM variable (suscritos) [,variable(suscritos)]...

Antes de poder utilizar una tabla de variables, el programa primero debe ejecutar una instrucción DIM para establecer las DIMensiones de dicha tabla (a menos que haya 11 elementos o menos). La instrucción DIM va seguida por el nombre de la tabla, que puede ser cualquier nombre legal de variable. A continuación, entre paréntesis, debe poner el número (o variable numérica) de elementos en cada dimensión. Una tabla con más de una dimensión se llama matriz. Puede Vd. utilizar cualquier número de dimensiones, pero tenga en cuenta que el conjunto de las variables que está Vd. creando ocupa un espacio en la memoria y es fácil agotarla. Para saber el número de variables creado con cada DIM, multiplique el número total de elementos de cada dimensión. (Cada tabla se inicia con el elemento 0).

NOTA: Las tablas enteras (de un solo dígito) ocupan 2/5 del espacio que ocupa una tabla en coma flotante.

EJEMPLO:

```
10 DIM A$ (40),B7(15),CC%(4,4,4)
```

41 elementos 16 elementos 125 elementos

Puede Vd. dimensionar más de una tabla con una instrucción DIM, separándolas por comas. Si el programa ejecuta una instrucción DIM para una tabla más de una vez, le dará el mensaje de error REDIM'D ARRAY. Es una buena práctica de programación colocar las instrucciones DIM cerca del inicio del programa.

DO (LOOP) WHILE (UNTIL EXIT)

DO [UNTIL argumento booleano WHILE argumento booleano]
instrucciones [EXIT]

LOOP [UNTIL argumento booleano WHILE argumento booleano]

(Un ejemplo de argumento booleano es A=1 ó H>=57).

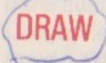
Efectúa las instrucciones entre la instrucción DO y LOOP. Si no hay ningún UNTIL o WHILE que modifique las instrucciones DO o LOOP, la ejecución de las instrucciones citadas continúa indefinidamente. Si se encuentra una instrucción EXIT en el cuerpo de un bucle DO, la ejecución es transferida a la primera instrucción tras la instrucción LOOP. Los bucles DO pueden ir anidados, siguiendo las reglas definidas para los bucles FOR...NEXT.

Si se utiliza el parámetro UNTIL, el bucle continúa efectuándose hasta que se satisface el argumento booleano (pasa a ser VERDADERO). El parámetro WHILE es, básicamente, el opuesto al parámetro UNTIL. El bucle continúa mientras el argumento booleano sea VERDADERO.

EJEMPLO:

DO UNTIL X=0 OR X=1

LOOP
DO WHILE A$ = " ": GET A$:LOOP

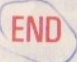


DRAW [origen del color #] [,ab,b1] [,TO a2, b2,] [...]

Con este comando puede Vd. dibujar puntos individuales, líneas y figuras. Vd. indica el origen del color (0-3) y los puntos de inicio (a1,b1) y final (a2,b2).

EJEMPLOS:

un punto:	DRAW 1, 100, 50	no se especifica punto final, por defecto se toman los valores a1, b1 para crear un punto.
lineas:	DRAW, 10,10,TO 100,60 DRAW TO 25,30	
figura:	DRAW, 10,10 TO 10,60 TO 100,60 TO 10,10	



END

Cuando el programa ejecuta una instrucción END, se detiene la ejecución. Puede utilizar el comando CONT para reiniciar el programa en la instrucción siguiente a la instrucción END.

FOR... TO... STEP

FOR variable = inicial TO valor final [STEP incremento]

Esta instrucción trabaja junto con la instrucción NEXT para ejecutar una sección del programa un cierto número de veces. Puede Vd. desear simplemente que su ordenador cuente hasta un número alto para que se detenga durante unos segundos. Puede necesitar también contar algo o hacer algo un cierto número de veces (por ejemplo, una impresión).

La variable del bucle es el valor que se suma o se sustrae durante la ejecución del bucle. Los valores inicial y final son los contadores inicial y final para la variable del bucle.

La lógica de la instrucción FOR es la siguiente. Primero, la variable del bucle es creada con el valor inicial. Luego, cuando el programa alcanza una línea con el comando NEXT, le suma el valor del incremento (por defecto = 1) y comprueba si es mayor que el valor final del bucle. Si no es mayor, la siguiente línea que se ejecuta es la siguiente a la instrucción FOR. Si la variable del bucle es mayor que el número de final de bucle, entonces la siguiente instrucción ejecutada es la que sigue a la instrucción NEXT. El valor de STEP puede ser positivo o negativo. Vea también la instrucción NEXT.

EJEMPLO:

```
10 FOR L = 1 TO 20
2O PRINT L
30 NEXT L
40 PRINT "BLACKJACK! L = "L
```

Este programa imprime en la pantalla los números de uno a veinte, seguidos por el mensaje BLACKJACK! L = 21.

El valor de final del bucle puede ir seguido por la palabra STEP y otro número o variable. En este caso, el valor siguiente a STEP es sumado, cada vez, en vez de uno. Esto le permite contar hacia atrás, por fracciones o de cualquier modo que lo necesite.

Puede Vd. colocar bucles unos dentro de otros. Esto es conocido como anidar bucles. Debe tener cuidado con los bucles anidados ya que el último bucle en empezar debe ser el primero en terminar.

EJEMPLOS DE BUCLES ANIDADOS

```
10 FOR L = 1 TO 100
20 FOR A = 5 TO 11 STEP 2
30 NEXT A
40 NEXT L
```

Este bucle está "anidado" dentro del mayor



GET

GET variable

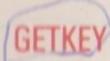
La instrucción GET es un modo de recoger datos desde el teclado, carácter por carácter. Cuando se ejecuta GET, el carácter tecleado es recibido, se devuelve en la variable y el programa continúa sin esperar ninguna otra pulsación. No es necesario pulsar la tecla **RETURN** y, de hecho, la tecla **RETURN** puede ser recogida mediante GET.

La palabra GET va seguida por un nombre de variable, usualmente de cadena. Si se utilizara una variable numérica y se pulsara una tecla no numérica, el programa se detendría dando un mensaje de error. La instrucción GET puede colocarse también dentro de un bucle, comprobando si el resultado es una cadena vacía, con lo cual el programa espera a que se pulse una tecla para continuar. La instrucción GETKEY puede también ser utilizada en este caso. Este comando solo puede ser ejecutado en un programa.

EJEMPLO:

```
10 GET A$: IF A$< > "A" THEN 10
```

Esta línea espera a que sea pulsada la tecla "R" para continuar.



GETKEY

GETKEY variables

La instrucción GETKEY es muy similar a la instrucción GET. A diferencia de esta, GETKEY espera a que el usuario teclee un carácter del teclado. Esto permite utilizarla fácilmente para esperar la pulsación de un único carácter.

Este comando solamente puede ejecutarse en un programa.

EJEMPLO:

```
10 GETKEY A$
```

Esta línea espera que sea pulsada una tecla. Tecleando cualquiera de ellas, el programa continuará con el carácter A$.

GET

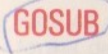
GET # número de fichero, variable

Usada con un periférico o fichero abierto previamente, lee un carácter cada vez. En otras palabras, actúa como la instrucción GET.

Este comando sólo puede ejecutarse en un programa.

EJEMPLO:

```
GET #1, A$
```



GOSUB

GOSUB línea #

Esta instrucción es como la instrucción GOTO, excepto que su COMMODORE 16 recuerda de donde ha venido. Cuando se encuentra una

línea con la instrucción RETURN, el programa salta de nuevo a la instrucción inmediatamente posterior a GOSUB. El objetivo de una instrucción GOSUB es llamar a una subrutina. Una subrutina es útil si hay una rutina en su programa que puede ser utilizada por varias porciones diferentes del mismo. En vez de duplicar la sección del programa una y otra vez, puede emplearla como una subrutina y acudir a ella desde las distintas partes del programa. Vea también la instrucción RETURN.

EJEMPLO:

20 GOSUB 800	significa ir a la subrutina que empieza en la línea 800 y ejecutarla

800 PRINT "EH, OIGA": RETURN

GOTO o GO TO

GOTO línea #

Tras la ejecución de una instrucción GOTO, la siguiente línea a ejecutar es la que sigue a la palabra GOTO. Cuando se usa en modo directo, GOTO línea # le permite iniciar la ejecución del programa en el número de línea especificado sin borrar las variables.

EJEMPLO:

```
10 PRINT "LA REPETICION ES LA MADRE DEL APRENDIZAJE"
20 GOTO 10
```

El GOTO de la línea 20 provoca que la línea 10 se esté ejecutando continuamente hasta que se pulse la tecla **RUN/STOP.**

GRAPHIC

GRAPHIC modo [opción de borrado]

Esta instrucción pone su COMMODORE 16 en uno de los 5 modos gráficos:

modo	descripción
0	texto normal
1	gráficos de alta resolución
2	gráficos de alta resolución, pantalla partida
3	gráficos multicolores
4	gráficos multicolores, pantalla partida

Cuando se ejecuta, GRAPHIC 1-4 asigna un área de 10K y el área de texto BASIC es colocada bajo esa área de alta resolución. Esta permanece asignada incluso cuando el usuario retorna al modo TEXTO (GRAPHIC 0). Si se da un 1 en el segundo argumento, la pantalla es también limpiada.

EJEMPLOS:

GRAPHIC 1,1	Selecciona el modo gráfico hi-res y limpia la pantalla
GRAPHIC 4,0	Selecciona el modo gráfico multicolor con un área para texto, sin limpiar la pantalla



GRAPHIC CLR

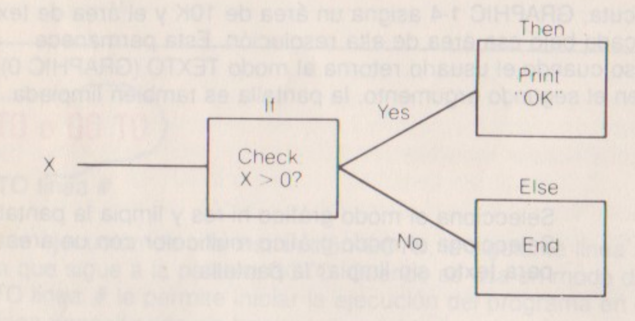
Esta es una forma de la instrucción GRAPHIC. En vez de especificar una forma de utilizar los gráficos, esta instrucción limpia las 10K de memoria asignada para gráficos y esa zona de memoria vuelve a estar disponible para el BASIC.

IF... THEN... :ELSE

IF expresión THEN cláusula [:ELSE cláusula alternativa]

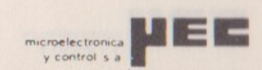
IF... THEN permite al ordenador analizar la expresión BASIC precedida por IF y tomar uno de los dos posibles cursos de acción. Si la expresión es cierta, la instrucción siguiente a THEN es ejecutada. Esta expresión puede ser una instrucción BASIC. Si la primera es falsa, el programa pasa directamente a la línea siguiente a menos que esté presente una cláusula ELSE. La expresión evaluada puede ser una variable o fórmula, en cuyo caso se considera cierta si es distinta de cero y falsa, si es cero. En la mayor parte de casos, hay una expresión utilizando operadores (=, <, >, <=, >=, < >, AND, OR, NOT).

La cláusula ELSE, si está presente, debe ir en la misma línea que IF... THEN. Cuando existe, se ejecuta cuando no lo hace la cláusula THEN. En otras palabras, se ejecuta cuando la expresión es falsa.



EJEMPLO:

50 IF X>0 THEN PRINT "OK": ELSE END	Comprueba el valor de X. Si es mayor que cero, se ejecuta la cláusula THEN y la ELSE no se ejecuta. Si X no es mayor que cero, se ejecuta la cláusula ELSE y no se ejecuta la cláusula THEN.





INPUT

INPUT ["pregunta";] lista de variables

La instrucción INPUT permite al ordenador pedir datos al usuario y los coloca en una o varias variables. El programa se detiene, imprime un interrogante (?) en la pantalla y espera a que se dé la respuesta y se pulse la tecla **RETURN.**

La palabra INPUT va seguida por el nombre de una variable o una lista de variables separadas por comas. Puede haber un mensaje entre comillas antes de la lista de variables. Si hay tal mensaje, debe llevar un punto y coma tras las comillas finales. Cuando hay más de una variable a introducir, deben separarse por comas cuando se escriban. Si no, el ordenador solicitará las variables restantes imprimiendo dos interrogantes (??). Si pulsa Vd. la tecla **RETURN** sin introducir ningún valor, las variables conservan los valores anteriores. Esta instrucción sólo puede ejecutarse en un programa.

EJEMPLO:

10 INPUT "CUAL ES SU NOMBRE",A$

20 INPUT "Y SU COLOR FAVORITO";B$

30 INPUT "CUAL ES LA VELOCIDAD DE VUELO DE UNA GOLONDRINA";A

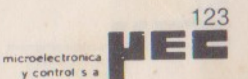
INPUT

INPUT# número de fichero, lista de variables

Trabaja igual que INPUT, pero cogiendo los datos de un fichero o unidad previamente abiertos. No permite texto. Este comando solamente puede ejecutarse en modo programa.

EJEMPLO:

1Ø INPUT# 2,A$,C,D$





LET

[LET] variable = expresión

La palabra LET es raramente utilizada en los programas ya que no es necesaria, pero la instrucción en sí es la base de los programas BASIC. Cada vez que se define o se le da un valor a una variable, está implícito LET. El nombre de la variable que va a recoger el resultado del cálculo va a la izquierda del signo igual y el número o fórmula va a la derecha.

EJEMPLO:

```
10 LET A=5

20 B=6

30 C=A*B+3

40 D$ = "HOLA"
```

LET está implícito (pero no es necesario) en las líneas 20, 30 y 40.

LOCATE

LOCATE coordenada x,coordenada y

El comando LOCATE le permite poner el cursor de pixels (PC) en cualquier lugar de la pantalla. El PC es la posición del punto de inicio del siguiente dibujo. A diferencia del cursor normal, Vd. no puede ver el PC, pero puede Vd. moverlo con el comando LOCATE. Por ejemplo:

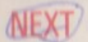
LOCATE 160,100

Posiciona el PC en el centro de la pantalla de alta resolución. No podrá ver nada hasta que no empiece a dibujar. Puede Vd. saber en donde se encuentra el PC en un momento determinado, utilizando la función RDOT (0) para conocer la coordenada x y RDOT(1) para la coordenada y. El orígen del color del punto en el PC puede averiguarse imprimiendo RDOT(2). (En todos los comandos de color en los que hay disponible la opción de seleccionar un color, puede seleccionar un valor de 0 a 3, correspondiente al fondo, caracteres, multicolor 1 ó multicolor 2 como orígen del color).

MONITOR

MONITOR

Este comando le lleva desde el BASIC al programa monitor de código de máquina. El monitor se utiliza para desarrollar, corregir y ejecutar programas en lenguaje máquina más fácilmente que con el BASIC. Vea la sección de comandos del monitor para más información. (Cuando esté en el monitor, tecleando "X" y pulsando **RETURN,** retornará al BASIC).



NEXT

NEXT [variable,..., variable]

La instrucción NEXT se utiliza con la instrucción FOR. Cuando el ordenador encuentra una instrucción NEXT, vuelve a la correspondiente instrucción FOR y comprueba el valor de la variable. (Vea la instrucción FOR para más detalles). Si el bucle ha terminado, la ejecución continúa con la instrucción posterior a NEXT. La palabra NEXT puede ser seguida por un nombre de una o varias variables, o puede no llevar nada. Si no lleva nada detrás, el último bucle iniciado es el que se completa. Si se dan las variables, se completan por orden de izquierda a derecha.

EJEMPLO:

```
10 FOR L = 1 TO 10:NEXT
20 FOR L = 1 TO 10:NEXT L
30 FOR L = 1 TO 10:FOR M = 1 TO 10:NEXT M,L
```

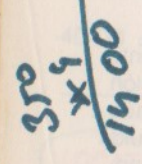
ON

ON expresión <GOTO/GOSUB> línea #1 [,línea #2,...]

Este comando puede convertir las instrucciones GOTO y GOSUB en versiones especiales de la instrucción IF. La palabra ON va seguida de una fórmula, luego GOTO o GOSUB y una lista de números de línea separados por comas. Si el resultado de calcular la fórmula (expresión) es 1, se ejecuta la primera línea de la lista. Si el resultado es 2, se ejecuta la segunda y así sucesivamente. Si el resultado es 0 o mayor que el número de líneas de la lista, la línea que se ejecuta es la siguiente a la instrucción ON. Si el número es negativo, dará como resultado el mensaje ILLEGAL QUANTITY ERROR.

EJEMPLO:

```
10 INPUT X:IF X<0 THEN 10
20 ON X GOTO 50, 30, 30, 70
25 PRINT "FRACASO": GOTO 10
30 PRINT "DEMASIADO ALTO": GOTO 10
50 PRINT "DEMASIADO BAJO": GOTO 10
70 END
```



Cuando X = 1, ON envía al programa a la primera línea de la lista (50).
Cuando X = 2, va a la segunda línea (30), etc.



OPEN

OPEN fichero # [, periférico # [.dir.sec. [,"nombre, fichero, tipo modo"]]]

La instrucción OPEN permite al COMMODORE 16 el acceso a periféricos tales como el grabador Datassette, los discos, una impresora e, incluso, la pantalla del monitor. La palabra OPEN va seguida de un número de fichero lógico, que es el número al que deberán referirse todas las otras instrucciones BASIC. Este número puede ir de 1 a 255. Hay habitualmente un segundo número tras el primero, llamado número de periférico. El periférico 0 es el teclado del COMMODORE 16, 1 es el grabador

Datassette (que se adjudica por defecto), 4 es la impresora, 8 es, usualmente, el disco. Se puede incluir un cero (0) delante del número de periférico (por ejemplo, 08 en vez de 8, ya que ambas formas son intercambiables por lo que concierne al COMMODORE 16). Es a menudo una buena idea utilizar el mismo número de fichero que el de periférico ya que, de ese modo, es más fácil recordar cual es cual. A continuación del segundo número puede ir un tercer número llamado dirección secundaria. En el caso del cassette, esta puede ser 0 para leer, 1 para escribir y 2 para escribir poniendo al final una marca de fin de cinta. En el caso del disco, el número se refiere al canal. En la impresora, la dirección secundaria se utiliza para activar el modo de impresión. Vea el Manual de Referencia del Programador del COMMODORE 16, o el manual de cada periférico específico para más información sobre las direcciones secundarias. Puede también haber una cadena tras el tercer número, que podría ser un comando para la unidad de disco o el nombre del fichero en cinta o en disco. El tipo y el modo se refieren solamente a ficheros en disco. (Los tipos de fichero son: prg, seq, rel y usr. Los modos son: read y write).

EJEMPLOS:

10 OPEN 3,3	Abre la pantalla como periférico.
10 OPEN 1,0	Abre el teclado como periférico.
20 OPEN 1,1,0, "UP"	Abre el cassette para leer el fichero llamado UP.
OPEN 4,4	Abre un canal para usar la impresora.
OPEN 15,8,15	Abre el canal de comandos.
5 OPEN 8,8,2, "TEST,SEQ,WRITE"	Crea un fichero de escritura secuencial en el disco.

Vea también las instrucciones CLOSE, CMD, GET# , INPUT# y PRINT# y las variables de sistema ST, DS y DS$.



PAINT

PAINT [color] [, [a,b] [,modo]]

color	(0-3). Por defecto es 1, color de los caracteres.
a.b	coordenadas de inicio (por defecto, PC).
modo	0 = llena un área definida por el origen del color seleccionado. 1 = llena un área definida por cualquier origen distinto al fondo.

El comando PAINT le permite rellenar un área de color. Llena el área alrededor del punto especificado hasta una frontera del mismo color (o cualquier color distinto del fondo, según el modo que haya elegido). La posición final del PC se encontrará en el punto de origen (a,b).

> **NOTA:** Si el punto de inicio es ya del color de origen que Vd. ha elegido (o de cualquier otro color distinto al fondo cuando utilice el modo 1), no habrá ningún cambio.

EJEMPLO:

10 CIRCLE, 160,100,65,50	dibuja una circunferencia
20 PAINT, 160,100	la rellena con el color

POKE

POKE dirección, valor

El comando POKE Le permite cambiar cualquier valor de la memoria RAM y de muchos de los registros de entrada/salida del COMMODORE 16. POKE va siempre seguido de dos números (o ecuaciones). El primer número es una dirección en la memoria del ordenador. Puede tener cualquier valor desde 0 a 65535. El segundo número es un valor de 0 a 255 que se colocará en la posición indicada, reemplazando cualquier valor que hubiera en ella previamente. Este comando puede utilizarse para controlarlo todo en la pantalla, desde colocar un carácter en una posición determinada hasta cambiar el color de la misma.

EJEMPLO:

10 POKE 16000,8	Pone un 8 en la posición 16000
20 POKE 16*1000,27	Pone 27 en la posición 16000

NOTA: PEEK, una función relacionada con POKE, está listada en el apartado de funciones.



PRINT

PRINT cadena a imprimir

La instrucción PRINT es la mayor instrucción de salida del BASIC. A pesar de que esta instrucción es la primera que la mayor parte de gente aprende a utilizar, tiene unas cuantas sutilezas en su manejo, que deben dominarse. La palabra PRINT puede ir seguida por cualquier combinación de estos elementos que se consideran la cadena a imprimir:

Caracteres entre comillas	(''líneas de texto'')
Nombres de variables	(A,B,A$,X$)
Funciones	(SIN(23).ABS(33))
Signos de puntuación	(;,)

Los caracteres entre comillas son llamados a menudo literales a causa de que se imprimen literalmente tal como aparecen. Los nombres de variables imprimen el valor que contienen (sea este un número o una cadena). Las funciones también imprimen sus valores. Los signos de puntuación se utilizan para conseguir un formato más claro en la pantalla. La coma divide la pantalla en cuatro columnas de datos, mientras que el punto y coma no deja ningún espacio entre datos. Cualquiera de estos dos signos puede ser utilizado como último símbolo de la instrucción. Esto tendrá como efecto que la siguiente instrucción PRINT actuará como si fuera una continuación de la anterior:

EJEMPLO:

	RESULTADO
10 PRINT "HOLA"	HOLA
20 A$ = "TU": PRINT "HOLA", A$	HOLA, TU
30 A = 4:B = 2:PRINT A + B	6
50 J = 41:PRINT J;:PRINT J-1	41 40
60 C = A + B:D = A:PRINT A;B;C,D	4 2 6 2

Vea también las funciones POS( ), SPC ( ) y TAB ( ).

PRINT#

PRINT# fichero# ,texto a imprimir

Hay algunas diferencias entre esta instrucción y PRINT. Primeramente, la palabra PRINT# va seguida de un número que se refiere al periférico o al fichero abierto previamente. El número va seguido de una coma y una lista de cosas a imprimir. El punto y coma espacia del mismo modo en que lo hacía en la instrucción PRINT. La coma enviará 10 espacios a la mayor parte de impresoras y puede ser utilizada como separador para ficheros en disco (vea la Guía de Referencia del Programador y el Manual de la Unidad de Discos). Algunos periféricos no pueden trabajar con TAB y SPC.

EJEMPLO:

```
100 PRINT#   , "HOLA!",A$,B$,
```

PRINT USING

PRINT | #número de fichero,| USING formato, lista de variables

Esta instrucción le permite definir el formato de la cadena y de los elementos numéricos que desea imprimir en la pantalla, impresora u otro periférico. Ponga el formato deseado entre comillas. Este es el formato de listado. Luego ponga un punto y coma y la lista de lo que desea imprimir en el formato de listado. La lista puede estar formada por variables o valores que desee lista, por ejemplo:

```
5 X = 32: y = 100.23: A$ = "GATO"
10 PRINT USING "$  ##.## ";13.25,X,Y
20 PRINT USING "   ###>#";"CBM",A$
```

Cuando ejecute Vd. esto, la linea 10 imprimirá:

$ 13.25$ 32.00$ *****

> imprime *****ya que el valor Y tiene 5 dígitos y no está de acuerdo con el formato de listado (tal como se explicó antes).

La línea 20 imprime esto:

CBM GATO   deja tres espacios antes de imprimir "CBM" tal como se definió en el formato de listado

CARACTER	NUMERICO	CADENA
Signo #	X	X
Más (+)	X	
Menos (-)	X	
Punto decimal (.)	X	
Coma (,)	X	
Signo dólar ($)	X	
Cuatro flechas (↑↑↑↑)	X	
Signo igual (=)		X
Signo mayor que (>)		X

El signo # reserva un espacio para un carácter en el campo de salida. Si el dato contiene más caracteres que signos # indicados en el formato, ocurrirá lo siguiente:

En un PRINT numérico, el campo entero es rellenado con asteriscos (*). No se imprime ningún número.

Por ejemplo:

10 PRINT USING " #### ";X

Para estos valores de X, este formato listará:

X = 12.34	12
X = 567.89	568
X = 123456	****

Para un PRINT ALFANUMERICO, la cadena de datos se trunca en los límites del campo. Solamente se imprimen tantos caracteres como signos haya en el formato. El truncamiento se efectúa por la derecha.

Los signos más (+) y menos (-) pueden utilizarse en la primera o en la última posición de un formato, pero no en ambas. El signo más se imprime si el número es positivo. El signo menos, si es negativo. Si utiliza Vd. el signo menos y el número es positivo, se imprimirá un carácter en blanco en la posición correspondiente al signo menos.

Si no utiliza ninguno de estos dos signos en un PRINT numérico, se imprimirá un signo menos delante del primer dígito o del signo dólar si el número es negativo y no se imprimirá ningún signo si es positivo. Esto

significa que puede Vd. imprimir un carácter más si el número es positivo. Si hay demasiados dígitos en el campo especificado por los signos # y +/-, entonces ocurrirá un desbordamiento y se llenará el campo con asteriscos(*).

Un símbolo de punto decimal (.) designa la posición del punto decimal en el número. Sólo puede Vd. especificar un punto decimal en un campo de formato. Si no lo especifica, el valor es redondeado al entero más próximo e impreso sin cifras decimales.

Cuando especifica Vd. un punto decimal, el número de dígitos anteriores al punto (incluyendo el signo menos si es negativo) no deben exceder el número de # anteriores al punto decimal. Si hay demasiados dígitos, se produce un desbordamiento y el campo es rellenado con asteriscos (*).

Una coma(,) le permite colocar comas en un campo numérico. La posición de la coma en el formato indica donde aparecerá la coma en el número listado. Solamente son impresas las comas en el interior del número. Las comas sin utilizar a la izquierda del primer dígito aparecen como carácter de llenado. Por lo menos un # debe preceder a la primera coma del campo.

Si especifica Vd. comas en un campo y el número es negativo, se imprimirá un signo menos como primer signo, incluso si esta posición estuviera especificada como coma.

EJEMPLOS:

CAMPO	EXPRESION	RESULTADO	COMENTARIO
##.##+	-.01	00.01-	Añadido el primer cero
##.#-	1	1.0	Añadido el cero del arrastre
####	- 100.5	-101	Redondeado sin decimales
####	-1000	****	Desbordamiento porque en el campo no caben cuatro dígitos y el signo menos
###	10	10.	Añadido punto decimal
#$##	1	$1	Signo $inicial

Un signo de dólar ($) significa que se imprimirá dicho signo en el número. Si quiere un signo de dólar flotante (que se imprima siempre delante del número), deberá especificar, por lo menos, un # antes del signo de dólar. Si especifica Vd. un signo de dólar sin ningún # delante, se imprimirá en la posición indicada en el formato.

Si especifica Vd. comas y/o signos más o menos en un campo de formato con un signo de dolar, el programa imprimirá la coma o el signo antes del signo de dólar.

Las cuatro flechas verticales (↑↑↑↑) se utilizan para especificar que el número debe ser impreso en notación exponencial. Debe utilizar # conjuntamente con ↑↑↑↑ para especificar la anchura del campo. ↑↑↑↑ debe aparecer detrás de # en el campo de formato.

Debe especificar cuatro flechas (↑↑↑↑) cuando quiere imprimir un número en formato exponencial (notación científica). Si especifica Vd. más de una pero menos de cuatro flechas, recibirá un mensaje de error de sintaxis. Si especifica más de cuatro flechas, solamente se utilizan las cuatro primeras. La quinta (y las subsiguientes) son interpretadas literalmente y no como símbolos de texto.

Un signo igual (=) se utiliza para centrar una cadena en el campo. Vd. especifica la anchura del campo mediante el número de caracteres (# y =) en el campo de formato. Si la cadena contiene menos caracteres que la anchura, del campo, es centrada en él. Si contiene más caracteres, los más a la derecha son truncados y la cadena llena el campo completo.

Un signo mayor que (>) se utiliza para justificar a la derecha una cadena en el campo. Vd. especifica la anchura del campo mediante el número de caracteres (# y >) en el campo de formato. Si la cadena contiene menos caracteres que la anchura del campo, es justificada. Si contiene más caracteres, los más a la derecha quedan truncados y la cadena llena completamente el campo.

PUDEF

PUDEF "de 1 a 4 caracteres"

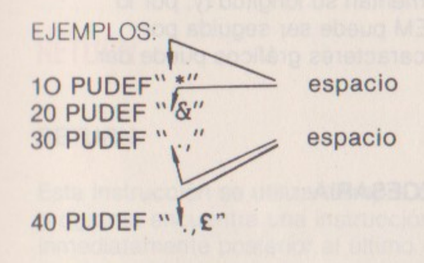
PUDEF le permite redefinir hasta 4 símbolos en las instrucciones PRINT USING. Puede Vd. cambiar blancos, comas, puntos decimales y signos de dólar por otros caracteres colocando el nuevo carácter en la posición correcta de la cadena de control PUDEF.

La posición 1 es el carácter de llenado. Por defecto es un blanco. Coloque aquí un nuevo carácter cuando quiera que ese carácter aparezca en vez de blancos.

La posición 3 es el punto decimal.

La posición 4 es el signo de dólar.

EJEMPLOS:



10 PUDEF "*"	espacio	Imprime * en vez de blancos
20 PUDEF "&"		Imprime & en vez de comas
30 PUDEF ".,"	espacio	Imprime puntos decimales en vez de comas y comas en vez de puntos decimales
40 PUDEF ".,£"		Imprime el signo de libra en vez del dólar, puntos decimales en vez de comas y comas en vez de puntos decimales.

READ

READ lista de variables

Esta instrucción se utiliza para pasar información de las instrucciones DATA a las variables, en donde van a ser utilizados los datos. La lista de variables de la instrucción DATA puede contener cadenas y variables numéricas. Debe llevar cuidado en evitar leer cadenas cuando la instrucción READ espera un número, ya que se producirá un mensaje de ERROR.

EJEMPLO:

READ A$, G$, Y

REM

REM Comentario

REM es simplemente una nota para quién quiera que lea un listado del programa. Puede explicar una sección del programa, dar información acerca del autor, etc. Las instrucciones REM no tienen ningún efecto en la ejecución del programa, excepto que aumentan su longitud (y, por lo tanto, lo hacen más lento). La palabra REM puede ser seguida por cualquier texto, aunque la utilización de caracteres gráficos puede dar peculiares resultados.

EJEMPLO:

```
10 NEXT X:REM ESTA LINEA ES INNECESARIA
```

RESTORE

RESTORE [línea #]

Cuando se ejecuta en un programa, el puntero de elementos de las instrucciones DATA que se acaban de leer, es restaurado al elemento inicial de la lista. Esto le dá la capacidad de releer la información. Si sigue a la instrucción una [línea #], el puntero es colocado en esa línea. De otro modo, es restaurado para leer la primera instrucción DATA del programa.

EJEMPLO:

```
RESTORE 200
```

RESUME

RESUME [línea # NEXT]

Se utiliza para volver a la ejecución tras corregir un error, sin argumentos, RESUME intenta ejecutar la linea en la que ocurrió el error. RESUME NEXT continúa la ejecución en la instrucción siguiente a aquella en la que ocurrió el error. RESUME línea # irá a la línea especificada e iniciará la ejecución en ella.

RETURN

RETURN

Esta instrucción se utiliza siempre con la instrucción GOSUB. Cuando el programa encuentra una instrucción RETURN, vuelve a la instrucción inmediatamente posterior al último comando GOSUB ejecutado. Si no se ha ejecutado ninguno previamente, se dará el mensaje RETURN WITHOUT GOSUB ERROR y se detendrá la ejecución del programa.

SCALE

SCALE <1/0>

La escala de los dibujos en modo de alta resolución y en multicolor puede ser cambiada con el comando SCALE. Introduciendo;

SCALE 1

se activa la escala. Las coordenadas pueden entonces ir desde 0 a 1023 en ambos ejes, x e y, en vez de la escala normal de valores, que es:

modo multicolor	X = 0 a 159	y = 0 a 199
modo de alta resolución	X = 0 a 319	y = 0 a 199

La escala se desactiva introduciendo "SCALE 0".

SCNLR

SCNLR

Limpia la pantalla actual, ya sea gráficos, texto o ambas (pantalla partida).

SOUND

SOUND voz # , frecuencia, duración

Esta instrucción produce un sonido utilizando una de las tres voces con un control de frecuencia del rango de 0 a 1023 para una duración de 0 a 65535 60avos de segundo

V	Voz
1	Voz 1 (tono)
2	Voz 2 (ruido)
3	Voz 2 (ruido blanco)

Si se desea un sonido en la voz N y el sonido anterior para la misma voz todavía está sonando, el BASIC espera a que el anterior haya terminado. SOUND con una duración de 0 es un caso especial. Causa que el BASIC desconecte el sonido de esa voz inmediatamente, sin importar el tiempo que reste para completarse dicho sonido. Vea la TABLA DE NOTAS MUSICALES en el apéndice para conocer los valores de frecuencia que corresponden a las notas reales.

EJEMPLO:

SOUND 2,800,3600 — Toca una nota utilizando la voz 2, en una frecuencia de 800, durante un minuto.

SSHAPE/GSHAPE

SSHAPE Y GSHAPE se utilizan para salvar y restaurar áreas rectangulares o multicolores de pantallas de alta resolución utilizando variables de cadena BASIC. El comando para salvar un área es:

SSHAPE variable de cadena, a1, b1 [,a2,b2]

variable de cadena	Nombre de la cadena en la que se van a salvar los datos
a1,b1	Coordenadas de la esquina
a2,b2	Coordenadas de la esquina opuesta (por defecto, el PC)

Ya que el límite de la longitud de las cadenas en BASIC es de 255 caracteres, el tamaño del área que puede Vd. salvar es limitado. El tamaño de la cadena requerida puede ser calculado utilizando una de las siguientes fórmulas:

L(mmc) = INIT ((ABS (a1-a2) + 1)/4 + .99) * (ABS(b1-b +) + 1) + 4

L(hr) = INIT ((ABS (a1-a2) + 1)/8 + .99) * (ABS(b1-b2) + 1) + 4

(mmc) se refiere al modo multicolor. (hr) es alta resolución.

La figura es salvada fila a fila. Los últimos cuatro bytes de la cadena contienen la longitud de la columna y de la fila menos uno (por ejemplo, ABS (a1-a2)) en el formato byte bajo/alto (si está con escala dividir las longitudes por 3.2 (x) y 5.12 (y)).

El comando para representar en la pantalla una figura previamente salvada es:

GSHAPE nombre de la variable [, [a,b] [, modo]]

variable	Contiene la figura a dibujar
a,b	Coordenada de la esquina superior izquierda en donde se va a dibujar la figura (a escala), por defecto es el PC
modo	Modo de colocación: 0: colocar la figura tal cual (por defecto) 1: colocarla en campo invertido 2: hacer OR de la figura con el área 3: hacer AND de la figura con el área 4: hacer XOR de la figura con el área

EJEMPLOS:

SSHAPE "VARIABLE$", 0,0 — Salva el área de pantalla desde la esquina superior izquierda hasta el lugar en que se encuentra el cursor, bajo el nombre de VARIABLE$.

GSHAPE "VARIABLE$" ,,,1 — Representa la figura VARIABLE $ con los colores invertidos, con el vértice superior izquierdo en la posición del cursor.

STOP

STOP

Esta instrucción detiene el programa. Da un mensaje, BREAK IN LINE #, en donde # es el número de línea conteniendo el STOP. El programa puede ser relanzado a partir de la instrucción siguiente a STOP utilizando el comando CONT. La instrucción STOP es utilizada habitualmente mientras se depura un programa.



SYS

SYS dirección

La palabra SYS va seguida por un número decimal o una variable numérica en el rango de 0 a 65535. El programa inicia la ejecución de un programa en código máquina que empieza en la posición de memoria indicada. Es similar a la función USR, pero no utiliza parámetro de paso. Vea la Guía de Referencia del Programador para información sobre programas en código máquina.

TRAP

TRAP [línea #]

Cuando se activa, TRAP intercepta todas las condiciones de error (inclyendo la tecla **RUN/STOP**) excepto "UNDEF'D STATEMENT ERROR". En caso de un error de ejecución, la bandera de error es activada y la ejecución se transfiere al número de línea indicado en la instrucción TRAP. El número de línea en el que ha ocurrido el error puede ser hallado utilizando la variable de sistema EL. La condición específica del error está contenida en la variable de sistema ER. La función de cadena ERR$ (ER) dá el mensaje de error correspondiente a cada condición de error ER.

> **NOTA:** Un error en una rutina TRAP no puede ser interceptado. La instrucción RESUME puede utilizarse para continuar la ejecución. TRAP sin argumento de línea # desconecta la intercepción de errores.

TRON

TRON

TRON se utiliza en la depuración de programas. Esta instrucción inicia el modo de rastreo. Cuando está Vd. en modo de rastreo, con cada instrucción ejecutada se imprime el número de línea.

TROFF

TROFF

Esta instrucción desactiva el modo de rastreo.

VOL

VOL nivel de volumen

Controla el nivel de volumen de los comandos SOUND. El VOLumen puede ser colocado a cualquier valor entre 0 y 8, siendo 8 el volumen máximo y 0, volumen desconectado. VOL afecta a ambas voces.

WAIT

WAIT dirección, valor 1 [, valor 2]

La instrucción WAIT se utiliza para detener el programa hasta que el contenido de una posición de memoria cambie de un modo específico. La dirección debe estar en el rango de 0 a 65535. El valor 1 y el valor 2 deben estar en el rango de 0 a 255.

El contenido de la posición de memoria primero es operado mediante OR exclusivo con el valor 2 (si existe) y, a continuación, operado mediante AND con el valor 1. Si el resultado es cero, el programa comprueba de nuevo la posición de memoria. Si es distinto de cero, el programa continúa con la siguiente instrucción.

Información adicional sobre instrucciones gráficas

Hay algunos conceptos que se aplican a todas las instrucciones de gráficos. Primero, el concepto del Cursor de Pixels (PC). El PC es similar al cursor del modo texto. Es la posición en que se va a dibujar el siguiente punto. A diferencia del cursor de texto, el PC es invisible. Todos los comandos de dibujo utilizan el PC. Además, el comando LOCATE le permite colocar el PC sin dibujar nada.

Cada vez que utilice coordenadas x,y en un comando de dibujo, puede utilizar coordenadas RELATIVAS. Las coordenadas relativas se basan en el valor actual del PC. Para utilizarlas, simplemente coloque un signo + ó -delante de ellas. Un signo más delante de la coordenada x significa que

el PC se mueve a la derecha. Un valor negativo, lo mueve a la izquierda. De manera similar, un signo menos en la coordenada y mueve el PC hacia arriba, mientras que un signo más, lo mueve hacia abajo. Por ejemplo:

LOCATE + 100,-25	mueve el PC 100 pixels a la derecha y 25 hacia arriba.
DRAW 1, + 10, + 10TO100,100	dibuja una línea desde la posición situada 10 pixels a la derecha y 10 hacia abajo del PC, hasta el punto 100,100.

Puede también especificar una distancia y un ángulo relativo al PC separando los dos parámetros mediante un punto y coma.

POR EJEMPLO:

LOCATE 50;45	mueve el PC desde su posición actual a una distancia de 50 puntos, en un ángulo de 45 grados.

FUNCIONES

Funciones numéricas

Las funciones numéricas son calificadas de ese modo porque dan un resultado numérico. Estas funciones abarcan desde el cálculo de funciones matemáticas hasta la especificación de una posición de pantalla. Las funciones numéricas siguen el formato:

FUNCION (argumento)

donde el argumento puede ser un valor numérico, una variable o una cadena.

ABS(X) (valor absoluto)

La función valor absoluto de la magnitud del argumento X.

ASC(X$)

Esta función da el código ASCII (número) de primer carácter de X$.

ATN(X) (arcotangente)

Da el ángulo, en radianes, cuya tangente es X.

COS (X) (coseno)

Da el valor del coseno de X, donde X es un ángulo medido en radianes.

DEC (cadena hexadecimal)

Da el valor decimal de una cadena hexadecimal <0 <cadena hexadecimal <FFFF>.

EJEMPLO:

N = DEC("F4")

EXP(X)

Da el valor de la constante matemática e (2.71828183) elevada a la potencia X.

FNxx(x)

Da el valor de la función xx definida por el usuario con una instrucción DEF FNxx.

INSTR (cadena 1, cadena 2 [,posición inicial])

Da la posición de la cadena 2 en la 1 tras la [posición inicial]. Por defecto, esta posición inicial es el principio de la cadena 2. Si no se ha encontrado correspondencia, se da el valor 0.

EJEMPLO:

PRINT INSTR ("EL GATO EN EL SOMBRERO", "GATO")

el resultado es 4, ya que GATO empieza en el cuarto carácter de la cadena 1.

INT (X) (entero)

Da la parte entera de X, eliminando todos los decimales. El resultado es siempre menor o igual a X. Así, cualquier número negativo con decimales pasa a ser el entero menor (por ejemplo, INT (-4.5) = -5).

Si la función INT se utiliza para redondear, debe utilizarse la forma INT(X + .5) o INIT (X-.5).

EJEMPLO:

X = INT(X*100 + .5)/100 Redondea al céntimo.

JOY (n)

Cuando n = 1 Posición del joystick #1
Cuando n = 2 Posición del joystick #2

El valor 128 o superior, significa que ha sido pulsado el botón de disparo. La dirección se indica del modo siguiente:

ARRIBA

disparo = 128 + 1

8 2

IZQUIERDA 7 0 3 DERECHA

6 4

5

ABAJO

EJEMPLO:

JOY(2) con valor de 135 dispara el joystick # 2 hacia la izquierda.

LOG (X) (logaritmo)

Da el logaritmo natural de X. El logaritmo natural es el logaritmo en base a (ver EXP(X)).

Para convertirlo a base 10, divídalo por LOG(10).

PEEK (X)

Esta función da el contenido de la posición de memoria X, donde X está en el rango de 0 a 65535. El resultado está comprendido entre 0 y 255. Se utiliza a menudo en conjunción con la instrucción POKE.

RCLR (N)

Da el color actual asignado al origen N <0 = < N = < 4). (0 = fondo, 1 = caracteres, 2 = multicolor, 1, 3 = multicolor 2, 4 = borde).

RDOT (N)

Pone la información sobre la posición actual del cursor de pixels (PC) en XPOS/YPOS.

N = 0 para XPOS
 1 para YPOS
 2 origen de color

RGR(X)

Da el modo gráfico actual (X es un falso argumento que puede tener cualquier valor).

RLUM(N)

Da el nivel de luminosidad asignado a la fuente de color N.

RND(X) (número aleatorio)

Esta función da un número aleatorio entre 0 y 1. Es muy utilizada en juegos, para simular un lanzamiento de dados y otros elementos de azar. También se utiliza en aplicaciones de estadística. El primer valor debería ser generado por la fórmula RND (-TI), para empezar de un modo diferente cada vez. Tras esto, el número X debería ser un 1, o cualquier número positivo. (X representa la semilla en la que se basa el número aleatorio). Si es cero, RND es re-semillado a partir del reloj de hardware cada vez que se utiliza RND. Un valor negativo de X activa el generador de números utilizando X como semilla y da una secuencia de números aleatorios. El uso del mismo número negativo para X da como resultado la misma serie de números. Un valor positivo da números basados en la semilla previa.

Para simular el lanzamiento de un dado, utilice la fórmula INT(RND(1)* 6 + 1). Primero, el número aleatorio es multiplicado por 6, lo que aumenta el valor de 0 a 6 (mayor de cero y menor de seis). Luego se le suma 1, pasando el rango a un valor entre 1 y 7. La función INT elimina los decimales, dejando como resultado un dígito de 1 a 6.

Para simular los dados, sume dos números obtenidos por la fórmula anterior.

EJEMPLO:

100 X=INIT(RND(1)*6)+INIT(RND)(1)*6)+2	Simula 2 dados
100 X=INT(RND(1)*1000)+1	Número de 1 a 1000
100 X=INT(RND(1)*150)+100	Número de 100 a 249

SGN(X) (signo)

Esta función da el signo, positivo, negativo o cero, de X. El resultado es +1 si X es positivo, 0 si es cero y -1 si es negativo.

SIN(X) (seno)

Esta es la función del seno trigonométrico. El resultado es el seno de X, donde X es un ángulo en radianes.

SQR(X) (raiz cuadrada)

Esta función da la raiz cuadrada de X, donde X es un número positivo o 0. Si es negativo, dará el resultado ILLEGAL QUANTITY ERROR.

TAN(X) (tangente)

Esto da la tangente de X, donde X es un ángulo en radianes.

USR(X)

Cuando se utiliza esta función, el programa salta a un programa en código máquina que se inicia en la posición contenida en las posiciones de

memoria 1281 y 1282. El parámetro X es traspasado al programa en lenguaje máquina a través del acumulador de coma flotante. Otro número es devuelto al BASIC a través de la variable de llamada. En otras palabras. Esto le permite intercambiar una variable entre el código máquina y el BASIC. Vea la Guía de Referencia del Programador para más detalles sobre esto y sobre la programación en código máquina.

VAL(X$)

Esta función convierte la cadena X$ en un número y es, esencialmente la operación inversa a STR$. La cadena es examinada a partir del carácter más a la izquierda, durante tantos caracteres como haya en un formato numérico reconocible. Si el COMMODORE 16 encuentra caracteres ilegales, solo se convierte la porción de cadena hasta ese punto.

EJEMPLO:

10 X = VAL("123.456")	X=123.456
10 X = VAL("3E03Z")	X=3000
10 X = VAL("12A13B")	X=12
10 X = VAL("RIUO17*")	X=0
10 X = VAL("-1.23.23.23")	X = -1.23

Funciones de cadena

Las funciones de cadena difieren de las funciones numéricas en el hecho de que dan como resultado caracteres, gráficos o números en una cadena (definida por comillas), en vez de un número.

CHR$(X)

Esta función da un carácter cuyo código ASCII es X.

ERR$(N)

Da la cadena que describe el error de condición N (ver TRAP).

HEX$(N)

Da una cadena de cuatro caracteres conteniendo la representación hexadecimal del valor N <0 <N <65535>.

LEFT$(X$,X)

Esto da una cadena conteniendo los X caracteres más a la izquierda de X$.

LEN(X$)

Da el número de caracteres (incluyendo espacios y otros símbolos) de la cadena X$.

MID$(X$,N,X)

Esto da una cadena conteniendo X caracteres, empezando por el carácter N de X$. MID$ puede también utilizarse a la izquierda de una instrucción de asignación como pseudo-variable. MID$ (variable de cadena, posición inicial, longitud) = cadena inicial.

Esta función reasigna los valores de las posiciones (posición inicial) durante (posición inicial + longitud) de la cadena original con los valores de los caracteres de la variable en las posiciones correspondientes. Por defecto, la longitud será la longitud de la variable y se producirá un error si la (posición inicial + longitud) es mayor que la longitud de la cadena original.

EJEMPLO:

```
10 A$ = "EL ULTIMO ADIOS":
20 PRINT A$
30 MID$(a$,4,6)= "PRIMER"
40 PRINT A$
```

RIGHT$(X$,X)

Esto da los X caracteres más a la derecha de X$.

STR$ (X)

Esto da una cadena que es idéntica a la versión impresa de X$.

EJEMPLO:

A$ = STR$(X)

Otras funciones

FRE(X)

Esta función da el número de bytes sin utilizar en la memoria. X es un falso argumento.

POS(X)

Esta función da el número de la columna (0-39) en la que empezará el siguiente argumento en la pantalla. X es un falso argumento.

SPC(X)

Esto se utiliza en las instrucciones PRINT para saltar X espacios. X puede tener cualquier valor desde 0 a 255.

TAB(X)

Esto se utiliza en las instrucciones PRINT. El siguiente elemento a imprimir lo será en la columna X. X puede tener un valor de 0 a 255.

π

El símbolo π, cuando se utiliza en una ecuación, tiene el valor 3.14159265.

VARIABLES Y OPERADORES

Variables

Su COMMODORE 16 utiliza tres tipos de variables en BASIC. Estas son: variables numéricas, enteras y de cadena (alfanuméricas).

Las **VARIABLES NUMERICAS,** también llamadas **variables de coma flotante,** pueden tener cualquier valor desde 10 elevado a -38 hasta 10 elevado a +38, con nueve dígitos de aproximación. Cuando un número sobrepasa los nueve dígitos que pueden representarse, como 10 elevado a -10 o 10 elevado a +10, el ordenador lo presenta en notación científica, con el número normalizado a 1 dígito con ocho decimales, seguido por la letra E y la potencia de 10 por la que se multiplica el número. Por ejemplo, el número 12345678901 se representa como 1.23456789E+10.

Las **VARIABLES ENTERAS** pueden ser utilizadas cuando el número va de +32767 a -32768 y este no lleva decimales. Una variable entera es un número como 5, 10 o -100. Los enteros utilizan menos espacio que las variables de coma flotante, cuando se utilizan en una tabla.

Las **VARIABLES DE CADENA** son las utilizadas para datos con caracteres que pueden ser números, letras y cualquier otro carácter que puede utilizar su COMMODORE 16. Un ejemplo de variable de cadena es "COMMODORE 16".

NOMBRES DE VARIABLES

Los nombres de las variables pueden consistir en una única letra, una letra seguida por un número o dos letras. Los nombres de variables pueden ser mayores de dos caracteres, pero sólo son significativos los dos primeros.

Una variable entera se especifica utilizando el signo de porcentaje (%) tras el nombre. Las variables de cadena llevan el signo de dólar ($) tras el nombre.

EJEMPLOS:

Nombres de variables numéricas: A, A5, BZ

Nombres de variables enteras: A%, A5%, BZ%

Nombres de variables de cadena: A$, A5$, BZ$

Las **TABLAS** son listas de variables con el mismo nombre que utilizan un número (o números) extra para especificar el elemento de la tabla. Las tablas se definen con las instrucción DIM y pueden ser de coma flotante, enteras o de cadena. El nombre de la variable de la tabla va seguido por un par de paréntesis ( ) incluyendo el número de la variable en el conjunto.

EJEMPLOS: A(7), BZ%(11), A$(87)

Las tablas pueden tener más de una dimensión. Una tabla de dos dimensiones puede ser visualizada como poseyendo filas y columnas, identificando el primer número la columna y el segundo, la fila (tal como se especifica una coordenada en un mapa).

EJEMPLOS: A(7,2), BZ%(2,3,4), Z$(3,2)

NOMBRES DE VARIABLE RESERVADOS

Hay siete nombres de variables que están reservados para uso del COMMODORE 16 y no se pueden utilizar para otro propósito. Estas son las variables DS, DS$, ER, EL, ST, TI y TI$. Tampoco puede Vd. utilizar, como nombre de variables, PALABRAS CLAVE como SRUN, RNEW o XLOAD.

ST es una variable de estado para entrada y salida (excepto pantalla normal y operaciones de teclado). El valor de ST depende del resultado de la última operación de entrada/salida efectuada. Una explicación más detallada de ST la encontrará en la Guía de Referencia del Programador —Serie 264, pero, en general, si el valor de ST es 0, la operación se ha efectuado correctamente.

TI y TI$ son variables relacionadas con el reloj de tiempo real incorporado a su COMMODORE 16. El reloj del sistema está actualizado cada 1/60 de segundo. Empieza desde 0 cuando se conecta el COMMODORE 16 y se restaura solamente al cambiar el valor de TI$. La variable TI le da el valor actual del reloj medido en 1/60 de segundo.

TI$ es una cadena que lee el valor del reloj de tiempo real como un reloj normal. Los primeros dos carácteres de TI$ contienen la hora, el tercero y el cuarto son los minutos y el quinto y sexto, los segundos. Esta variable puede ser inicializada con cualquier valor (mientras sean caracteres numéricos) y será automáticamente actualizado como en un reloj.

EJEMPLO:

TI$ = ''101530'' inicializada el reloj a las 10 h. 15 m. 30 s. (A.M.).

El valor del reloj se pierde cuando se desconecta el ordenador. Empieza desde cero al conectarlo de nuevo y se restaura a cero cuando el valor del reloj excede de 235959 (23h. 59m. 59s.)

La variable DS lee el canal de comandos de disco y da el resultado actual de la unidad. Para explicarlo claramente, imprime DS$. Estas variables de

estado se utilizan tras una operación en disco, tal como DLOAD o DSAVE, para encontrar el motivo por el que parpadea la luz roja de la unidad.

ER, EL y ERR$ son variables utilizadas en las rutinas de rastreo de errores. Unicamente son útiles en un programa. ER da el último error encontrado desde que se empezo a ejecutar el programa. EL es la línea en donde ocurrió el error. ERR$ es una función que permite al programa imprimir uno de los mensajes de error BASIC. PRINT ERR$(ER) imprime el mensaje de error apropiado.

OPERADORES BASIC

Los operadores ARITMETICOS incluyen los siguientes signos:

+ adición

- substración

* multiplicación

/ división

↑ elevación a una potencia (exponenciación)

En una línea que contenga más de un operador, las operaciones se realizan en un orden preestablecido. Si se utilizan varios operadores juntos, el ordenador asigna las prioridades del modo siguiente:

Primero, exponenciación, luego, multiplicación y división y, por último, adición y substracción. Si dos operaciones tienen la misma prioridad, los cálculos se efectúan por orden, de izquierda a derecha. Si desea que las operaciones se efectúen en diferente orden, el BASIC del COMMODORE 16 le permite dar una prioridad más alta a un cálculo colocándolo entre paréntesis. Las operaciones encerradas entre paréntesis se calcularán antes que cualquier otra. Debe asegurarse de que sus ecuaciones contengan el mismo número de paréntesis de apertura que de cierre, de lo contrario, al ejecutar el programa, le dará un mensaje de SYNTAX ERROR.

Hay también operadores para igualdades y desigualdades, llamados operadores RELATIVOS. Los operadores aritméticos tienen también prioridad sobre los operadores relativos.

=	igual a
<	menor que
>	mayor que
<= ó =<	menor o igual
>= ó =>	mayor o igual
< > ó > <	no igual a

Finalmente, hay tres operadores LOGICOS, que tienen una prioridad menor que los aritméticos y los relativos:

AND
OR
NOT

Estos se utilizan muy a menudo para unir fórmulas múltiples en una instrucción IF... THEN. Cuando se utilizan con operadores aritméticos, se evalúan en último lugar (por ejemplo, tras + y -).

EJEMPLOS:

IF A = B AND C = D THEN 100	requiere que A = B y C = D sean ciertos
IF A = B OR C = D THEN 100	permite que uno de los dos sea cierto
A = 5:B = 4 PRINT A = B	representa el valor 0
A = 5;B = 4: PRINT A>B	representa el valor 1
PRINT 123 AND 15:PRINT 5 OR 7	representa 11 y 7

Abreviaturas BASIC y Tabla de Referencia

PALABRA CLAVE	ABREVIATURA			TIPO
ABS	a	**SHIFT**	B	función numérica
ASC	a	**SHIFT**	S	función numérica
ATN	a	**SHIFT**	T	función numérica
AUTO	a	**SHIFT**	U	comando
BACKUP	b	**SHIFT**	A	comando
BOX	b	**SHIFT**	O	instrucción
CHAR	ch	**SHIFT**	A	instrucción
CHR$	c	**SHIFT**	H	función de cadena
CIRCLE	c	**SHIFT**	I	instrucción
CLOSE	cl	**SHIFT**	O	instrucción
CLR	c	**SHIFT**	L	instrucción
CMD	c	**SHIFT**	M	instrucción
COLLECT	col	**SHIFT**	L	comando
COLOR	co	**SHIFT**	L	instrucción
CONT	c	**SHIFT**	O	comando
COPY	co	**SHIFT**	P	comando
COS	ninguna			función numérica
DATA	d	**SHIFT**	A	instrucción
DEC	ninguna			función numérica
DEF IN	d	**SHIFT**	E	instrucción
DELETE	de	**SHIFT**	L	comando
DIM	d	**SHIFT**	I	instrucción
DIRECTORY	di	**SHIFT**	R	comando
DLOAD	d	**SHIFT**	L	comando
DO	ninguna			instrucción
DRAW	d	**SHIFT**	R	instrucción
DSAVE	d	**SHIFT**	S	comando
END	e	**SHIFT**	N	instrucción
ERR$	e	**SHIFT**	R	función de cadena
EXP	e	**SHIFT**	X	función numérica
FOR	f	**SHIFT**	O	instrucción
FRE	f	**SHIFT**	R	función numérica
GET	g	**SHIFT**	E	instrucción
GETKEY	getk	**SHIFT**	E	instrucción
GET #	ninguna			instrucción
GOSUB	go	**SHIFT**	S	instrucción
GOTO	g	**SHIFT**	O	instrucción
GRAPHIC	g	**SHIFT**	R	instrucción
GSHAPE	g	**SHIFT**	S	instrucción
HEADER	he	**SHIFT**	A	comando
HEX$	h	**SHIFT**	E	función de cadena
IF...GOTO	ninguna			instrucción

PALABRA CLAVE	ABREVIATURA	TIPO
IF... THEN... ELSE	ninguna	instrucción
INPUT	ninguna	instrucción
INPUT #	i **SHIFT** N	instrucción
INSTR	in **SHIFT** S	función numérica
INT	ninguna	función numérica
JOY	j **SHIFT** O	función numérica
KEY	k **SHIFT** E	comando
LEFT$	le **SHIFT** F	función de cadena
LEN	ninguna	función numérica
LET	l **SHIFT** E	instrucción
LIST	l **SHIFT** I	comando
LOAD	l **SHIFT** O	comando
LOCATE	lo **SHIFT** C	instrucción
LOG	ninguna	función numérica
LOOP	lo **SHIFT** O	instrucción
MID$	m **SHIFT** I	función de cadena
MONITOR	m **SHIFT** O	instrucción
NEW	ninguna	comando
NEXT	n **SHIFT** E	instrucción
ON... GOSUB	on...	
	go **SHIFT** S	instrucción
ON...GOTO	o	
	n...g **SHIFT** O	instrucción
OPEN	o **SHIFT** P	instrucción
PAINT	p **SHIFT** A	instrucción
PEEK	p **SHIFT** E	función numérica
POS	ninguna	función numérica
PRINT	?	instrucción
PRINT #	p **SHIFT** R	instrucción
PRINT USING	?us **SHIFT** I	instrucción
PUDEF	p **SHIFT** U	instrucción
RCLR	r **SHIFT** C	función numérica
RDOT	r **SHIFT** D	función numérica
READ	r **SHIFT** E	instrucción
REM	ninguna	instrucción
RENAME	re **SHIFT** N	comando
RENUMBER	ren **SHIFT** U	comando
RESTORE	re **SHIFT** S	instrucción
RESUME	res **SHIFT** U	instrucción
RETURN	re **SHIFT** T	instrucción
RGR	r **SHIFT** G	función mumérica
RIGT$	r **SHIFT** I	función de cadena
RLUM	r **SHIFT** L	función numérica
RND	r **SHIFT** N	función numérica
RUN	r **SHIFT** U	comando



PALABRA CLAVE	ABREVIATURA	TIPO
SAVE	s **SHIFT** A	comando
SCALE	sc **SHIFT** A	instrucción
SCNCLR	s **SHIFT** C	instrucción
SCRATCH	sc **SHIFT** R	comando
SGN	s **SHIFT** G	función numérica
SIN	a **SHIFT** I	función numérica
SOUND	s **SHIFT** 8O	instrucción
SPC<	s **SHIFT** P	función especial
SQR	s **SHIFT** Q	función numérica
SSHAPE	s **SHIFT** S	instrucción
STatus	ninguna	variable mumérica
STOP	s **SHIFT** T	instrucción
STR$	st **SHIFT** R	función de cadena
SYS	s **SHIFT** Y	instrucción
TAB<	t **SHIFT** A	función especial
TAN	ninguna	función numérica
TI	ninguna	variable numérica
TI$	ninguna	variable de cadena
TRAP	t **SHIFT** R	instrucción
TROFF	tro **SHIFT** F	instrucción
TRON	tr **SHIFT** O	instrucción
UNTIL	u **SHIFT** N	instrucción
USR	u **SHIFT** S	función especial
VAL	ninguna	función numérica
VERIFY	v **SHIFT** E	comando
VOL	v **SHIFT** O	instrucción
WAIT	w **SHIFT** A	instrucción
WHILE	w **SHIFT** H	instrucción

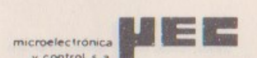


APENDICES

APENDICE A
MENSAJES DE ERROR

Estos mensajes de error se imprimen por el BASIC. También puede Vd. imprimirlos utilizando la función ERR$( ). El número de error solamente se refiere al número asignado al error para utilizarlo con dicha función.

ERROR #	NOMBRE DEL ERROR	
1	**TOO MANY FILES**	Demasiados ficheros. Hay un límite de diez ficheros abiertos simultáneamente.
2	**FILE OPEN**	Fichero abierto. Se ha intentado abrir un fichero con el número de otro fichero ya abierto.
3	**FILE NOT OPEN**	Fichero no abierto. El número de fichero especificado en una instrucción e/s debe ser abierto antes de utilizarlo.
4	**FILE NOT FOUND**	Fichero no encontrado. No existe el fichero de ese nombre en el disco.
5	**DEVICE NOT PRESENT**	Periférico no presente. El periférico requerido no está disponible. (Sin tensión o no conectado).
6	**NOT INPUT FILE**	No es fichero de entrada. Se ha intentado leer datos de un fichero definido para salida.
7	**NOT OUTPUT FILE**	No es fichero de salida. Se ha intentado enviar datos a un fichero definido para entrada.
8	**MISSING FILE NAME**	Falta nombre de fichero. Un OPEN, LOAD o SAVE en unidad de disco, generalmente necesita un nombre de fichero.
9	**ILLEGAL DEVICE NUMBER**	Número de periférico ilegal. Se ha intentado utilizar un periférico impropiamente (SAVE a la pantalla, etc).



10	**NEXT WITHOUT FOR**	NEXT sin FOR. Hay algún bucle mal anidado o un nombre de variable en una instrucción NEXT que no se corresponde con la de FOR.
11	**SYNTAX**	Sintaxis. Hay una instrucción irreconocible por el BASIC. Puede ser a causa de algún paréntesis de menos o de más, palabras clave mal deletreadas, etc.
12	**REURN WITHOUT GOSUB**	RETURN sin GOSUB. Se ha encontrado una instrucción RETURN sin haberse encontrado antes ningún GOSUB.
13	**OUT OF DATA**	Fin de datos. Se ha encontrado una instrucción READ y no hay datos para leer.
14	**ILLEGAL QUANTITY**	Cantidad legal. Se ha utilizado como argumento de una función, un número fuera de rango.
15	**OVERFLOW**	Desbordamiento. El resultado de una operación es mayor que el mayor número permitido (1.701411833E + 38).
16	**OUT OF MEMORY**	Memoria terminada. O no hay más espacio para el programa y las variables, o hay demasiadas DO, FOR o GOSUB en funcionamiento.
17	**UNDEF'D STATEMENT**	Instrucción no definida. El número de línea indicado no existe en el programa.
18	**BAD SUBSCRIPT**	Subscrito erróneo. El programa intenta utilizar un elemento de una tabla fuera del rango dado en la instrucción DIM.
19	**REDIM'D ARRAY**	Tabla redimensionada. Una tabla solo se puede dimensionar una vez. Si se hace referencia a una tabla

		antes de dimensionarla, se hace de modo automático (a 10).
20	**DIVISION BY ZERO**	División por cero. La división por cero no está permitida.
21	**ILLEGAL DIRECT**	Direccionamiento ilegal en modo directo. Las instrucciones INPUT o GET sólo están permitidas en un programa.
22	**TYPE MISMATCH**	Error de tecleo. Esto ocurre cuando se utiliza un número en vez de una cadena, o viceversa.
23	**STRING TOO LONG**	Cadena demasiado larga. Una cadena puede contener hasta 255 caracteres.
24	**FILE DATA**	Datos de fichero. Datos mal leídos desde un fichero en cinta.
25	**FORMULA TOO COMPLEX**	Fórmula demasiado compleja. Simplifique la expresión (sepárela en dos partes o utilice menos).
26	**CAN'T CONTINUE**	No se puede continuar. El comando CONT no funciona si el programa no se estaba ejecutando, si hay un error o si se ha corregido una línea.
27	**UNDEF'D FUNCTION**	Función no definida. Se ha hecho referencia a una función del usuario que no ha sido definida.
28	**VERIFY**	Verificación. El programa en cinta o en disco no coincide con el que hay en la memoria.
29	**LOAD**	Carga. Hubo un problema en la carga. Inténtelo de nuevo.
30	**BREAK**	Corte. Se ha pulsado la tecla STOP para detener la ejecución del programa.
31	**CAN'T RESUME**	No se puede seguir. Se ha encontrado una instrucción RESUME sin estar activo la instrucción TRAP.
32	**LOOP NOT FOUND**	No se ha encontrado LOOP. El programa ha encontrado una instrucción DO y no puede encontrar la correspondiente LOOP.
33	**LOOP WITHOUT DO**	LOOP sin DO. Se ha encontrado una instrucción LOOP sin estar en acción una instrucción DO.
34	**DIRECT MODE ONLY**	Sólo en modo directo. Este comando sólo se permite en modo directo, no desde un programa.
35	**NO GRAPHICS AREA**	No hay área de gráficos. Se ha encontrado un comando para crear gráficos (DRAW, BOX, etc.) antes de que se haya ejecutado el comando GRAPHIC.
36	**BAD DISK**	Disco estropeado. Ha fallado un intento de formatear un diskette porque el método de formateado rápido (sin ID) se ha intentado en un diskette no formateado previamente, o porque el diskette es defectuoso.

APENDICE B
MENSAJES DE ERROR DE DISCO

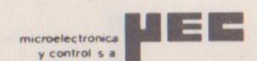
Estos mensajes de error son dados a través de las variables reservadas DS y DS$.

NOTA: Los números de mensaje de error menores de 20 deben ignorarse, a excepción de 01, que da información acerca del número de ficheros borrados con el comando SCRATCH.

20	**READ ERROR** (no se ha encontrado cabecera de bloque)	Error de lectura. El controlador de disco es incapaz de localizar la cabecera del bloque de datos solicitado. La causa puede ser un número de sector ilegal o que la cabecera haya sido destruida.
21	**READ ERROR** (no hay sincronismo de carácter)	Error de lectura. El controlador de disco es incapaz de detectar una marca de sincronismo en la pista deseada. La causa puede ser una mala alineación de las cabezas de lectura/escritura, que no esté presente el diskette o que esté mal colocado o no haya sido formateado. Puede indicar también un fallo de hardware.
22	**READ ERROR** (bloque de datos no presente)	Error de lectura. El controlador de disco ha intentado leer o verificar un bloque de datos que no estaba correctamente grabado. Este mensaje de error ocurre en conjunción con el comando BLOCK e indica una pista o sector ilegales.
23	**READ ERROR** (error de comprobación en un bloque de datos)	Error de lectura. Este mensaje indica que hay un error en uno o más bloques de datos. Estos han sido leídos en la memoria DOS, pero en la comprobación ha habido un error. Este mensaje puede indicar también problemas básicos.
24	**READ ERROR** (error de decodificación del byte)	Error de lectura. El dato o la cabecera han sido leídos en la memoria DOS pero se ha creado un error de hardware debido a una configuración de bits no válida en el byte de datos. Este mensaje puede también indicar problemas básicos.
25	**WRITE ERROR** (error de escritura-verificación)	Error de escritura. Este mensaje se genera cuando el controlador detecta una discrepancia entre los datos y los datos en la memoria DOS.
26	**WRITE PROTECT ON**	Protección de escritura. Este mensaje se genera cuando el controlador ha intentado grabar un bloque de datos con el interruptor de protección de escritura pulsado. Esto viene causado por la utilización de un diskette con una tira de protección de escritura sobre la muesca.
27	**READ ERROR** (error de comprobación en cabecera)	Error de lectura. El controlador ha detectado un error en la cabecera del bloque de datos solicitado. El bloque no ha sido leído en la memoria DOS. Este mensaje puede también indicar problemas básicos.
28	**WRITE ERROR** (bloque de datos largo)	Error de escritura. Se intenta detectar la marca de sincronismo de la siguiente cabecera tras leer un bloque de datos. Si la marca de sincronismo no aparece en un tiempo determinado, se genera el mensaje de error. Este es causado por un mal formato de diskette (los datos se extienden hasta el bloque siguiente) o por un fallo de hardware.
29	**DISK ID MISMATCH**	No coincide identificación de disco. Este mensaje se genera cuando el controlador intenta acceder a un

diskette que no ha sido inicializado. El mensaje puede aparecer también si el diskette ha sido mal formateado.

30	**SYNTAX ERROR** (sintaxis general)	Error de sintaxis. El DOS no puede interpretar el comando enviado al canal de comandos. Esto es causado por un número ilegal de nombres de fichero, o porque las configuraciones han sido utilizadas de modo ilegal. Por ejemplo, pueden aparecer dos nombres de fichero a la izquierda del comando COPY.
31	**SYNTAX ERROR** (comando no válido)	Error de sintaxis. El DOS no reconoce el comando. Este debe empezar en la primera posición.
32	**SYNTAX ERROR** (comando no válido)	Error de sintaxis. El comando es superior a 58 caracteres.
33	**SYNTAX ERROR** (nombre de fichero no válido)	Error de sintaxis. Configuración mal utilizada en un comando OPEN o SAVE.
34	**SYNTAX ERROR** (no se ha dado fichero)	Error de sintaxis. El nombre del fichero ha sido omitido en un comando o el DOS no lo reconoce como tal. Han sido olvidados los dos puntos (:) en el comando.
39	**SYNTAX ERROR** (comando no válido)	Error de sintaxis. Este error puede aparecer si el comando (dirección secundaria 15) no es reconocido por el DOS.
50	**RECORD NOT PRESENT**	Registro no presente. La lectura del disco ha rebasado el último registro, en un comando INPUT # o GET #. Este mensaje se presentará tras posicionarse en un registro posterior al fin de fichero en un fichero relativo. Si se intenta expandir el fichero añadiendo el nuevo registro (con un comando PRINT #), el mensaje de error se puede ignorar. INPUT o GET no se deben intentar, tras la detección de este error sin una reposición previa.
51	**OVERFLOW IN RECORD**	Desbordamiento de registro. La instrucción PRINT # excede los límites del registro. Los datos son truncados. Ya que el retorno de carro que marca el final del registro cuenta en la longitud, este mensaje aparecerá si los caracteres totales del registro (incluyendo el retorno de carro exceden el tamaño definido).
52	**FILE TOO LARGE**	Fichero demasiado largo. La posición del registro en un fichero relativo indica que ocurrirá un desbordamiento.
60	**WRITE FILE OPEN**	Fichero de escritura abierto. Este mensaje se generará cuando un fichero de escritura que no haya sido cerrado e intente ser abierto para lectura.
61	**FILE NOT OPEN**	Fichero no abierto. Este mensaje se genera cuando se accede a un fichero que no ha sido abierto en el DOS. Algunas veces, en ese caso, no se genera mensaje. La petición simplemente se ignora.
62	**FILE NOT FOUND**	Fichero no encontrado. El fichero solicitado no existe en la unidad indicada.
63	**FILE EXISTS**	Existe fichero. El nombre del fichero que se está creando ya existe en el diskette.
64	**FILE TYPE MISMATCH**	Tipo de fichero erróneo. El tipo de fichero en el directorio no coincide con el tipo de fichero solicitado.



65	**NO BLOCK**	No hay bloques. Este mensaje ocurre en conjunción con el comando B-A. Indica que el bloque que se va a asignar ya ha sido asignado previamente. Los parámetros indican la pista y el sector disponible con el siguiente número mayor. Si los parámetros son cero (0), todos los bloques con número superior están en uso.
66	**ILLEGAL TRACK AND SECTOR**	Pista y sector ilegales. El DOS ha intentado acceder a una pista o bloque que no existe en el formato utilizado. Puede indicar un problema en la lectura del puntero del siguiente bloque.
67	**ILLEGAL SYSTEM T OR S**	Pista o sector ilegal del sistema. Este mensaje especial de error indica que hay una pista o un sector del sistema que es ilegal.
70	**NO CHANNEL** (disponible)	No hay canal. El canal pedido no esta disponible o están todos ellos en uso. Pueden ser abiertos por el DOS un máximo de cinco ficheros secuenciales al mismo tiempo. (Los canales de acceso directo, 6).
71	**DIRECTORY ERROR**	Error en el directorio. El BAM no coincide con el contador interno. Hay un problema en la asignación del BAM o el BAM ha sido sobreescrito en la memoria DOS. Para corregir este problema reinicialice el diskette para restaurar el BAM en la memoria. Algunos ficheros activos pueden ser terminados por la acción correctiva. NOTA: BAM = Mapa de bloques disponibles.
72	**DISK FULL**	Disco lleno. O se han terminado los bloques disponibles en el diskette o el directorio está al límite de su capacidad. Se envía DISK FULL cuando quedan disponibles en el 1541 2 bloques para permitir que se cierre el fichero actual.
73	**DOS MISMATCH (73, CBM DOS V2.6 1541)**	DOS incompatible. Los DOS 1 y 2 son compatibles en lectura pero no en escritura. Los discos son intercambiables en lectura pero un disco que haya sido formateado en una versión no puede usarse para grabar en la otra versión ya que el formato es diferente. Este error aparecerá cada vez que se intente escribir en un disco con un formato no compatible. (Hay disponible una rutina de utilidad para ayudarle en la conversión de un formato a otro.) Este mensaje también puede darse al conectar el sistema;
74	**DRIVE NOT READY**	Unidad no preparada. Se ha intentado el acceso a una unidad de discos floppy sin haber un diskette presente.



APENDICE C
FUNCIONES MATEMATICAS DERIVADAS

Las funciones que no son intrínsecas al BASIC 3.5 pueden calcularse del modo siguiente:

FUNCION	EQUIVALENTE BASIC
SECANTE	SEC(X) = 1/COS(X)
COSECANTE	CSC(X) = 1/SIN(X)
COTANGENTE	COT(X) = 1/TAN(X)
ARCO SENO	ARCSIN(X) = ATN(X/SQR(-X*X + 1))
ARCO SENO	ARCCOS(X) = -ATN(X/SQR(-X*X + 1)) + π/2
ARCO SECANTE	ARCSEC(X) = ATN(X/SQR(X*X-1))
ARCO SECANTE	ARCCS(X) = ATN(X/SQR(X*X-1)) + (SGN(X)-1*π/2)
ARCO COTANGENTE	ARCCOT(X) = ATN(X) + π/2
SENO HIPERBOLICO	SINH(X) = EXP(X)-EXP(-X))/2
COSENO HIPERBOLICO	COSH(X) = (EXP(X) + EXP(-X))/2
TANGENTE HIPERBOLICA	TANH(X) = EXP(-X)/(EXP(X) + EXP(-X))*2 + 1
SECANTE HIPERBOLICA	SECH(X) = 2/(EXP(X) + EXP(-X))
COSECANTE HIPERBOLICA	CSH(X) = 2/(EXP(X)-EXP(-X))
COTANGENTE HIPERBOLICA	COTH(X) = EXP(-X)/(EXP(X)-EXP(-X)*2 + 1
ARCO SENO HIPERBOLICO	ARCSINH(X) = LOG(X + SQR(X*X + 1))
ARCO COSENO HIPERBOLICO	ARCCOSH(X) = LOG(X + SQR(X*X-1))
ARCO TANGENTE HIPERB.	ARCTANH(X) = LOG((1 + X)/(1-X))/2
ARCO SECANTE HIPERB.	ARCSECH(X) = LOG((SQR(-X*X + 1) + 1/X)
ARCO COSECANTE HIPERB.	ARCCSCH(X) = LOG(SGN(X)*SQR (X*X + 1)/X)
ARCO COTANGENTE HIPERB.	ARCCOTH(X) = LOG((X + 1)/(X-1))/2

APENDICE D
TABLA DE NOTAS MUSICALES

NOTA	VALOR DEL REGISTRO DE SONIDO	FRECUENCIA (HZ)
LA	7	110
SI	118	123.5
DO	169	130.8
RE	262	146.8
MI	345	164.7
FA	383	174.5
SOL	453	195.9
LA	516	220.2
SI	571	246.9
DO	596	261.4
RE	643	293.6
MI	685	330
FA	704	349.6
SOL	739	392.5
LA	770	440.4
SI	798	494.9
DO	810	522.7
RE	834	588.7
MI	854	658
FA	864	699
SOL	881	782.2
LA	897	880.7
SI	911	989.9
DO	917	1045
RE	929	1177
MI	939	1316
FA	944	1398
SOL	953	1575

La tabla anterior contiene los valores del registro de sonido de cuatro octavas. Los valores se utilizan como segundo parámetro en el comando SOUND. Para utilizar la primera nota de la tabla (A - valor del registro de sonido,7) utilice el 7 como segundo número tras el comando SOUND-SOUND 1,7,30.

Utilice la fórmula siguiente para encontrar los valores del registro de sonido para frecuencias distintas a las de la tabla:

VALOR DEL REGISTRO DE SONIDO = 1024-(111860.781/FRECUENCIA)

Los valores de la tabla y de la fórmula anterior son para televisores NTSC. este es el sistema de televisión utilizado en Estados Unidos y Canadá. Si se encuentra Vd. en un país donde la televisión standard es en sistema PAL, (que es el caso de España) deberá utilizar la siguiente fórmula para calcular los nuevos valores del registro de sonido de toda la tabla:

VALOR DE REGISTRO DE SONIDO = 1024-(111840.45/Frecuencia)

APENDICE E

CODIGOS DE PANTALLA

La tabla siguiente lista todos los caracteres incorporados a su COMMODORE 16. Muestra qué números deben ponerse mediante POKE en la memoria de pantalla (posiciones 3072 a 4085) para obtener el carácter deseado. (Recuerde activar la memoria de color -2084 a 3071). También indica qué carácter corresponde al valor contenido al hacer PEEK desde la pantalla.

Hay disponibles dos juegos de caracteres, pero solamente uno a la vez. Esto significa que Vd. no puede tener en la pantalla caracteres de uno y de otro juego al mismo tiempo. Los juegos son alternados pulsando simultáneamente las teclas **SHIFT** y .

Desde el BASIC, PRINT CHR$(142) le pondrá en el modo mayúsculas/gráficos y PRINT CHR$(14) pasará al modo mayúsculas/minúsculas.

Cualquier número de la tabla puede ser también representado en INVERSO. El código de carácter inverso puede obtenerse sumando 128 a los valores mostrados.

JUEGO 1	JUEGO 2	POKE	JUEGO 1	JUEGO 2	POKE	JUEGO 1	JUEGO 2	POKE
@		0	T	t	20	(		40
A	a	1	U	u	21	)		41
B	b	2	V	v	22	*		42
C	c	3	W	w	23	+		43
D	d	4	X	x	24	,		44
E	e	5	Y	y	25	-		45
F	f	6	Z	z	26	.		46
G	g	7	[		27	/		47
H	h	8	£		28	0		48
I	i	9	]		29	1		49
J	j	10	↑		30	2		50
K	k	11	←		31	3		51
L	l	12	ESPACIO		32	4		52
M	m	13	!		33	5		53
N	n	14	"		34	6		54
O	o	15	#		35	7		55
P	p	16	$		36	8		56
Q	q	17	%		37	9		57
R	r	18	&		38	:		58
S	s	19	'		39	;		59

JUEGO 1	JUEGO 2	POKE	JUEGO 1	JUEGO 2	POKE	JUEGO 1	JUEGO 2	POKE
<		60		T	84			108
=		61		U	85			109
>		62		V	86			110
?		63		W	87			111
		64		X	88			112
	A	65		Y	89			113
	B	66		Z	90			114
	C	67			91			115
	D	68			92			116
	E	69			93			117
	F	70			94			118
	G	71			95			119
	H	72	ESPACIO		96			120
	I	73			97			121
	J	74			98		✓	122
	K	75			99			123
	L	76			100			124
	M	77			101			125
	N	78			102			126
	O	79			103			127
	P	80			104			
	Q	81			105			
	R	82			106			
	S	83			107			

Los Códigos de 128 a 255 son imágenes inversas de los 0 a 127

APENDICE F
CODIGOS ASCII Y CHR$

Este apéndice muestra los caracteres que aparecerán si hace Vd. PRINT CHR$(X), para todos los posibles valores de X. También le muestra los valores obtenidos haciendo PRINT ASC("X"), donde X es cualquier carácter que Vd. escriba. Esto es útil para evaluar el carácter recibido en una instrucción GET, pasar al modo mayúsculas/minúsculas e imprimir comandos basados en caracteres (como el cambiar a modo mayúsculas/minúsculas) que no pueden ponerse entre comillas.

PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$
	0	↑	17	"	34	3	51
	1	RVS ON	18	#	35	4	52
	2	CLR HOME	19	$	36	5	53
	3	INST DEL	20	%	37	6	54
	4		21	&	38	7	55
WHT	5		22	'	39	8	56
	6		23	(	40	9	57
	7		24	)	41	:	58
INHABILITA SHIFT CCC	8		25	*	42	;	59
HABILITA SHIFT CCC	9		26	+	43	<	60
	10	ESCAPE	27	,	44	=	61
	11	RED	28	−	45	>	62
	12	→	29	.	46	?	63
RETURN	13	GRN	30	/	47	@	64
CAMBIAR A MINUSCULAS MAYUSCULAS	14	BLU	31	0	48	A	65
	15	SPACE	32	1	49	B	66
	16	!	33	2	50	C	67

PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$	PRINTS	CHR$
D	68		97		126		155
E	69		98		127	PUR	156
F	70		99		128		157
G	71		100		129	YEL	158
H	72		101	FLASH ON	130	CYN	159
I	73		102	FLASH OFF	131	SPACE	160
J	74		103		132		161
K	75		104	f1	133		162
L	76		105	f3	134		163
M	77		106	f5	135		164
N	78		107	f7	136		165
O	79		108	f2	137		166
P	80		109	f4	138		167
Q	81		110	f6	139		168
R	82		111	HELP	140		169
S	83		112	SHIFT RETURN	141		170
T	84		113	CAMBIAR A MAYUSCULAS GRAFICOS	142		171
U	85		114		143		172
V	86		115	BLK	144		173
W	87		116	↓	145		174
X	88		117	RVS OFF	146		175
Y	89		118	CLR HOME	147		176
Z	90		119	INST DEL	148		177
[	91		120		149		178
£	92		121		150		179
]	93		122		151		180
↑	94		123		152		181
←	95		124		153		182
	96		125		154		183

APENDICE G

- Curso de introducción al Basic: Parte I (aparición en breve)
- Curso de introducción al Basic: Parte II (aparición en breve)
- Guía de referencia del programador de la serie 264 (aparición en breve)

INDICE ALFABETICO